

BTS SNIR	Document ressource
Lycée Jean Rostand Villepinte	Gestion des fichiers et des utilisateurs sous Debian

Gestion des fichiers et des utilisateurs

Table des matières

I	Présentation.....	1
I.1	Objectif du document.....	1
I.2	Votre compte.....	2
II	Accès aux fichiers.....	2
II.1	Déplacement dans des répertoires.....	2
II.2	Création d'un répertoire.....	3
II.3	Effacement d'un répertoire.....	4
II.4	Les jokers.....	4
II.5	Copie de fichiers ou de répertoires.....	4
II.6	Déplacement de fichiers ou de répertoires.....	5
II.7	Suppression de fichiers ou de répertoires.....	5
II.8	Modifier ou créer un fichier.....	6
II.9	Édition d'un fichier.....	6
II.10	Visualisation du contenu d'un fichier.....	6
II.11	Visualisation de la liste des fichiers d'un répertoire.....	6
II.12	Recherche de fichiers.....	7
II.13	Droits et propriétaires.....	8
II.14	Modification du propriétaire.....	8
II.15	Modification des droits.....	9
II.16	Modification du groupe propriétaire.....	10
III	Les groupes.....	10
III.1	Ajout d'un groupe.....	10
III.2	Suppression d'un groupe.....	10
III.3	Modification d'un groupe.....	10
IV	Les comptes utilisateur.....	11
IV.1	Ajout d'un utilisateur.....	11
IV.2	Configuration d'un utilisateur.....	11
IV.3	Changement du mot de passe d'un utilisateur.....	12
IV.4	Changement de l'utilisateur courant.....	12
IV.5	Commande userdel.....	12
IV.6	Modification des paramètres d'un utilisateur.....	13
IV.7	Gestion des groupes d'un utilisateur.....	13
IV.8	Affichage des groupes d'un utilisateur.....	13

I Présentation

I.1 Objectif du document

Ce document a pour objectif de voir comment on peut gérer des utilisateurs sous Linux.

Quand vous avez installé votre système Linux, il vous a demandé un nom pour un compte. Vous avez donc créé un premier compte ou utilisateur.

De base il existe un certain nombre de comptes déjà créés. Vous pouvez en avoir la liste en tapant dans un terminal :

```
more /etc/passwd
```

Le premier compte à voir est « **root** ». Ce compte est très particulier, il s'agit de l'administrateur, il a tous les droits et il faut éviter de l'utiliser car en cas d'erreur, vous pouvez détruire votre système.

BTS SNIR	Document ressource
Lycée Jean Rostand Villepinte	Gestion des fichiers et des utilisateurs sous Debian

I.2 Votre compte

Le compte que vous avez créé, a des droits plus restreints. Donc nous allons nous intéresser à ce compte, par exemple le compte snir. Si on affiche le fichier `/etc/passwd`, la dernière ligne contient la description de ce compte :

```
snir:x:1000:1000:snir,,,:/home/snir:/bin/bash
```

Cette ligne contient le nom du compte, le numéro correspondant à ce compte (1000, son identifiant), le numéro correspondant à son groupe (1000), une description non complètement rempli dans notre cas, son répertoire (`/home/snir`) et son interpréteur de commande (`/bin/bash`).

Voyons plus en détail ces informations.

- x Le nom du compte ne pose pas de problème, en général, il s'agit de votre nom.
- x L'identifiant du compte est une référence unique.
- x Le groupe est une entité qui va permettre de regrouper plusieurs comptes avec des caractéristiques communes.
- x La description comprend : le nom complet de l'utilisateur, son numéro de bureau, son numéro de téléphone professionnel et son numéro de téléphone personnel.
- x Son répertoire est son répertoire de travail. Dès qu'il se connecte, il se trouve dans ce répertoire. Il a tous les droits sur ce répertoire, il peut donc lire, écrire, modifier des fichiers et créer, modifier, supprimer des répertoires. Si vous avez changé de répertoire, vous pouvez y retourner en tapant la commande « `cd` ».
- x L'interpréteur de commande est le logiciel qui va comprendre et exécuter les commandes que vous allez entrer dans votre terminal. Celui que nous utilisons généralement est l'interpréteur `bash` qui se trouve dans le répertoire `/bin`.

Voyons quels sont les droits de cet utilisateur. A priori, il n'en a que peu et ne peut bien sûr pas exécuter des commandes système. Mais comme il s'agit du compte de celui qui a installé le système donc son administrateur, il aura comme même le droit d'exécuter des commandes système. Il devra ajouter avant la commande le mot « **sudo** », par exemple :

```
sudo ifconfig
```

Si vous voulez plus d'explications sur cette commande, vous devez lire le document sur cette commande.

II Accès aux fichiers

Nous allons voir les spécificités des fichiers de notre utilisateur `snir`. Tout d'abord, nous allons nous déplacer dans les répertoires grâce à la commande **cd**. Nous verrons ensuite comment déplacer, créer, modifier des fichiers ou des répertoires grâce aux commandes **cp**, **mv**, **rm**, **mkdir**, **rmdir** et **touch**. Puis nous allons visualiser les fichiers et répertoires contenus dans un répertoire grâce à la commande **ls**. Nous étudierons enfin des commandes pour changer les droits à des fichiers ou répertoires grâce aux commandes **chmod**, **chown**, **chgrp**.

II.1 Déplacement dans des répertoires

Nous avons vu que nous sommes de base dans le répertoire « `/home/snir` » (notre utilisateur s'appelle `snir`). Si ce n'est pas le cas, la commande **cd** sans aucun argument permet de retourner dans ce répertoire.

Sous Linux, les répertoires forment une arborescence comme sous Windows. Quand on parle du répertoire « `/home/snir` », cela signifie que nous avons l'arborescence suivante :

<code>/</code>	racine de tous les répertoires (il n'y a pas de C:,D: comme sous Windows)
<code>home/</code>	répertoire contenant tous les répertoires des utilisateurs
<code>snir</code>	répertoire de notre utilisateur <code>snir</code>

Si vous voulez savoir dans quel répertoire vous êtes, vous pouvez le voir dans le prompt (message affiché avant d'entrer sa commande ou bien grâce à la commande **pwd** :

BTS SNIR	Document ressource
Lycée Jean Rostand Villepinte	Gestion des fichiers et des utilisateurs sous Debian

```
snir@serveurDebian:/var/log$ pwd
/var/log
snir@serveurDebian:/var/log$ cd
snir@serveurDebian:~$ pwd
/home/snir
```

Le caractère ~ qui apparaît dans le prompt indique que le répertoire en cours est le répertoire de l'utilisateur courant.

Il existe deux techniques pour se déplacer dans les répertoires :

- x en relatif, c'est-à-dire par rapport au répertoire dans lequel on est ;
- x en absolu, sans tenir du répertoire actuel.

En absolu, il suffit de faire suivre la commande cd suivie par un espace puis le chemin complet, par exemple :

```
snir@serveurDebian:~$ cd /var/log
snir@serveurDebian:/var/log$ cd /home/snir
snir@serveurDebian:~$ pwd
/home/snir
```

En relatif, vous indiquez votre chemin par rapport au chemin actuel. Par exemple pour aller dans le répertoire parent, il faut taper « .. » :

```
snir@serveurDebian:/var/log$ pwd
/var/log
snir@serveurDebian:/var/log$ cd ..
snir@serveurDebian:/var$ pwd
/var
```

Si maintenant, nous voulons retourner dans le répertoire « /var/log », il suffit de donner le chemin « log » :

```
snir@serveurDebian:/var$ cd log
snir@serveurDebian:/var/log$ pwd
/var/log
```

« .. » désigne le répertoire parent, « . » désigne le répertoire en cours. Donc nous pouvons refaire la même commande en tapant « ./log » au lieu de « log » :

```
snir@serveurDebian:/var$ cd ./log
snir@serveurDebian:/var/log$ pwd
/var/log
```

Prenons d'autres exemples moins évidents :

```
snir@serveurDebian:~$ cd travail/projet
snir@serveurDebian:~/travail/projet$ cd ../TP
snir@serveurDebian:~/travail/TP$ cd ../../programmes/reseau/
snir@serveurDebian:~/programmes/reseau$ _
```

II.2 Création d'un répertoire

La commande pour la création d'un répertoire est **mkdir**.

Syntaxe :

```
mkdir [options] répertoires
```

Description :

La commande mkdir (make directory) permet de créer un ou plusieurs répertoires.

Options les plus courantes :

- m Définit le mode d'accès au répertoire (voir la commande chmod pour plus de renseignements).
- p Permet de créer les répertoires pères spécifiés.

Exemple : création du répertoire /home/bmaheux

```
mkdir /home/bmaheux
```

Autre exemple : création du répertoire travail/projet dans le répertoire courant

BTS SNIR	Document ressource
Lycée Jean Rostand Villepinte	Gestion des fichiers et des utilisateurs sous Debian

```
snir@serveurDebian:~$ mkdir travail/projet
mkdir: impossible de créer le répertoire « travail/projet »: Aucun fichier ou dossier de ce type
snir@serveurDebian:~$ mkdir travail
snir@serveurDebian:~$ mkdir travail/projet
snir@serveurDebian:~$ _
```

La première commande a été refusée car le répertoire travail devait d'abord être créé sinon il fallait ajouter l'option « -p » :

```
mkdir -p travail/projet
```

II.3 Effacement d'un répertoire

La commande pour l'effacement d'un répertoire est **rmdir**.

Syntaxe :

```
rmdir [options] [répertoires]
```

Description :

Supprime le ou les répertoires spécifiés (les répertoires doivent être vides).

Option la plus courante :

-p Supprime les répertoires pères spécifiés.

Exemple : supprime le répertoire /home/bmaheux (attention : il doit être vide)

```
rmdir /home/bmaheux
```

Remarque : Si le répertoire n'est pas vide, il faut utiliser la commande rm avec l'option -R :

```
rm -R /home/bmaheux
```

II.4 Les jokers

Si vous voulez supprimer, par exemple tous les fichiers qui ont pour extension « .jpg », vous pouvez par exemple supprimer chaque fichier l'un après l'autre ou bien utiliser le joker « * » qui correspond à des caractères quelconques :

```
rm *.jpg
```

Vous pouvez aussi supprimer tous les fichiers qui se commencent par projet :

```
rm projet*
```

Les fichiers projet1, projet2 et projetFutur sont effacés. Mais si vous ne voulez que supprimer les deux premiers fichiers, vous pouvez utiliser le joker « ? » qui correspond à un **seul** caractère quelconque :

```
rm projet?
```

II.5 Copie de fichiers ou de répertoires

La commande pour la copie de fichiers ou de répertoires est **cp**.

Syntaxe :

```
cp [options] source destination
```

Description :

Copie un fichier source vers un autre fichier (qui sera écrasé si ce fichier existe). Copie également un ensemble de fichiers vers un répertoire existant.

Options les plus courantes :

-f Efface les fichiers cibles existants.

-i Demande confirmation avant suppression d'un fichier.

-p Préserve les informations concernant le propriétaire, le groupe, les droits d'accès, et la date.

-r Effectue une copie récursive.

-u Ne copie pas un fichier qui en écraserait un plus récent.

Exemple : copier tous les fichiers avec l'extension txt du répertoire courant dans le répertoire /home/bmaheux

BTS SNIR	Document ressource
Lycée Jean Rostand Villepinte	Gestion des fichiers et des utilisateurs sous Debian

```
cp *.txt /home/bmaheux
```

Exemple : copier le répertoire programmes dans le répertoire temp

```
cp -r ./programmes ./temp
```

Remarque : si le répertoire temp existe déjà, le répertoire programmes est copié avec son contenu. Dans le cas contraire, le répertoire temp est créé puis le contenu du répertoire programmes est copié dans ce répertoire.

II.6 Déplacement de fichiers ou de répertoires

La commande pour le déplacement de fichiers ou de répertoires est **mv**.

Syntaxe :

```
mv [options] source cible
```

Description :

Déplace ou renomme des fichiers ou répertoires. Suivant la source et la cible, voici le résultat de la commande :

Source	Cible	Résultat
Fichier	Nouveau Fichier	Renomme le fichier
Fichier	Fichier existant	Écrase le fichier cible avec le fichier source
Répertoire	Nouveau Répertoire	Renomme le répertoire
Répertoire	Répertoire existant	Déplace le répertoire source dans le répertoire cible
Fichiers	Répertoire existant	Déplace les fichiers dans le répertoire cible

Options les plus courantes :

- f Supprime les fichiers sans confirmation.
- i Demande confirmation avant la suppression d'un fichier.
- u Ne déplace pas un fichier qui en écraserait un plus récent.

Exemples :

x déplace le fichier tmp.txt du répertoire courant dans le répertoire /home/bmaheux/

```
mv tmp.txt /home/bmaheux/
```

x renomme le fichier tmp.txt en bmaheux.txt

```
mv tmp.txt bmaheux.txt
```

II.7 Suppression de fichiers ou de répertoires

La commande est **rm**.

Syntaxe :

```
rm [options] fichiers
```

Description :

Efface le ou les fichiers spécifiés.

Options les plus courantes :

- d Efface le répertoire spécifié (même s'il n'est pas vide).
- i Demande confirmation avant d'effacer le ou les fichiers.
- r, -R Efface récursivement les sous-répertoires.

Exemples :

x supprime tous les fichiers dont l'extension est obj du répertoire courant

```
rm *.obj
```

x supprime le répertoire /home/bmaheux et tous les fichiers et sous-répertoires qu'il contient

```
rm -R /home/bmaheux
```

BTS SNIR	Document ressource
Lycée Jean Rostand Villepinte	Gestion des fichiers et des utilisateurs sous Debian

```
snir@serveurDebian:~$ pwd
/home/snir
snir@serveurDebian:~$ cd rep1/rep2/rep3
snir@serveurDebian:~/rep1/rep2/rep3$ pwd
/home/snir/rep1/rep2/rep3
snir@serveurDebian:~/rep1/rep2/rep3$ cd ../../../../rep4/rep5
snir@serveurDebian:~/rep4/rep5$ pwd
/home/snir/rep4/rep5
```

II.8 Modifier ou créer un fichier

La commande **touch** permet de créer un nouveau fichier ou de modifier sa date de dernière modification.

Syntaxe :

```
touch [-acm][-r référence [-tdate] [--] fichier
```

Description :

touch modifie la date d'accès et la date de modification de chaque fichier indiqué. Ces horodatages sont remplacés par la date et l'heure actuelles, sauf si l'option -r est utilisée.

Options les plus courantes :

- a Modifier l'heure du dernier accès au fichier
- c Ne pas créer les fichiers n'existant pas.
- m Changer l'heure de dernière modification du fichier
- t date Utiliser la date indiquée à la place de la date actuelle. L'argument est un nombre décimal de la forme : [[SS]AA]MMJJhhmm[.ss]

Exemple : création du fichier `essai.txt` si il n'existe pas ou sa date de modification est l'heure et la date actuelles.

```
touch essai.txt
```

II.9 Édition d'un fichier

Pour éditer ou créer un fichier texte, il faut utiliser la commande **nano**.

Syntaxe :

```
nano <chemin>
```

L'aide en ligne affichée en bas de l'écran lors de l'exécution de nano donne toutes les indications pour utiliser cet éditeur de texte.

II.10 Visualisation du contenu d'un fichier

Les commandes sur le contenu de fichiers sont :

x Afficher les 10 dernières ligne d'un fichier :

```
tail <chemin>
```

x Afficher le contenu d'un fichier :

```
cat <chemin>
```

```
more <chemin>
```

II.11 Visualisation de la liste des fichiers d'un répertoire

Pour visualiser le contenu d'un répertoire, il suffit d'entrer la commande ls :

```
snir@serveurDebian:~$ ls
fltk-1.3.3-docs-pdf.tar.gz  php-5.6.5.tar.bz2
```

Les fichiers apparaissent avec une couleur et les répertoires avec une autre.

Pour avoir le contenu d'un autre répertoire, il faut entrer le nom du répertoire en relatif ou en absolu comme pour la commande cd :

```
snir@serveurDebian:~$ cd rep1
snir@serveurDebian:~/rep1$ ls ..
fltk-1.3.3-docs-pdf.tar.gz  php-5.6.5.tar.bz2
```

BTS SNIR	Document ressource
Lycée Jean Rostand Villepinte	Gestion des fichiers et des utilisateurs sous Debian

Vous pouvez sélectionner certains fichiers grâce au caractère * qui va remplacer dans la liste des fichiers dont le nom comporte n'importe quels caractères. Par exemple si on précise les fichiers « t*e » on obtient la liste des fichiers commençant par un t et finissant par un e.

Une option très intéressante est « -l » qui donne des informations importantes telles que la taille et les droits des fichiers.

Syntaxe :

```
ls [options] [noms]
```

Description :

Affiche le contenu d'un répertoire (répertoire courant si non spécifié). Si un nom de fichier est spécifié, affiche la liste des fichiers correspondants.

Options les plus courantes :

- a Affiche tous les fichiers des répertoires, fichiers commençant par un '.' inclus.
- d Affiche les répertoires, sans lister leur contenu.
- l Affichage au format long (inclus le type du fichier, les droits d'accès, le propriétaire, la taille, ...)
- s Affiche la taille de chaque fichier (en blocs de 1 Ko).
- t Affiche le contenu des répertoires en fonction de la date (fichiers les plus récents présentés en premier).
- u Affiche le contenu des répertoires en fonction de la date de dernière utilisation (accès) au fichier.
- R Affiche le contenu de façon récursive (répertoires et sous-répertoires).
- S Affichage en fonction de la taille (du plus gros au plus petit).
- X Affichage en fonction de l'extension. Les fichiers sans extension sont affichés en premier.

Exemple : affiche la liste des fichiers avec l'extension txt du répertoire /home/snir en format long (avec le type, les droits, les propriétaires et la taille).

```
root@serveurDebian:~# ls -l /home/snir/*.txt
-rw-r--r-- 1 snir snir 8 juin 23 14:37 /home/snir/essai.txt
-rw-r--r-- 1 snir snir 8 juin 23 14:38 /home/snir/test.txt
```

II.12 Recherche de fichiers

La commande qui permet de rechercher un fichier dans un répertoire, est : find.

Syntaxe :

```
find [chemins] [conditions]
```

Description :

Effectue une recherche de fichiers suivant les conditions spécifiées. Le chemin par défaut est le répertoire courant.

Options les plus courantes :

- mtime n Effectue une recherche de fichier modifié dans les plus ou moins (+ ou -) n 24 heures.
- name nom Effectue une recherche de fichier dont le nom correspond à celui spécifié.
- print Affiche entièrement les noms de fichiers et de répertoires qui correspondent.
- type type Effectue une recherche de fichier dont le type est spécifié :
 - x b Fichier spécial en mode bloc
 - x c Fichier spécial en mode caractère
 - x d Répertoire
 - x p Fifo ou pipe nommé
 - x f Fichier
 - x l Lien symbolique

BTS SNIR	Document ressource
Lycée Jean Rostand Villepinte	Gestion des fichiers et des utilisateurs sous Debian

x s Socket

-size n[bckw] Effectue une recherche de fichier dont la taille est n : b pour blocs, c pour octets, k pour kilo-octets et w pour mots.

-user utilisateur Effectue une recherche de fichier appartenant à l'utilisateur spécifié.

Exemple : recherche de tous les fichiers ou répertoire ayant pour nom bmaheux à partir du répertoire racine :

```
root@debian0:~# find / -name bmaheux
/home/bmaheux
```

II.13 Droits et propriétaires

Chaque fichier ou répertoire appartient à un utilisateur et à un groupe utilisateur. Le propriétaire est celui qui crée le fichier. Chaque utilisateur appartient au moins à un groupe, le groupe propriétaire. Le groupe propriétaire est a priori le groupe du propriétaire.

Les droits sur les fichiers dépendent de l'utilisateur. Mais voyons d'abord quels sont les droits.

Il existe trois types de permissions sur un fichier :

x droit de lecture « r »,

x droit d'écriture (ou de modification) « w »,

x droit d'exécution « x ».

Les types d'accès de fichiers suivant le type de fichier sont :

Accès	Signification sur fichier	Signification sur répertoire
r	Affichage du CONTENU du fichier	Lecture du CONTENU du répertoire (ls)
w	Altération du CONTENU du fichier	Altération du CONTENU (= effacement fichiers)
x	Exécution du fichier	Accès à ce répertoire (cd sur lui)

La commande « ls -l » permet de connaître les droits d'accès données sur 10 caractères : un caractère donnant le type de fichier, trois caractères pour chaque sorte d'utilisateur : propriétaire, groupe propriétaire et les autres.

Ces trois caractères indiquent respectivement si le fichier est accessible en lecture, en écriture et en exécution (un tiret indique que non, la lettre correspondante que oui).

Exemple :

```
-rwxr-xr-l pierre system 120 Mar 12 16 :10 exemple
```

Le fichier « exemple » est : normal, tous les accès autorisés pour le propriétaire pierre, lecture et exécution pour les utilisateurs du groupe system et juste lecture pour les autres.

Il est possible de modifier les droits pour le propriétaire, le groupe propriétaire et les autres. Également il est possible de modifier le propriétaire, le groupe propriétaire.

Pour effectuer ces commandes, il faut être le propriétaire du fichier/répertoire ou avoir les droits du super-utilisateur (root).

II.14 Modification du propriétaire

La commande est **chown**.

Syntaxe :

```
chown [options] propriétaire[:groupe] fichiers
```

Description :

chown modifie l'utilisateur et/ou le groupe propriétaire de chacun des fichiers indiqués, en fonction du premier argument rencontré sur la ligne de commande, et qui ne soit pas une option. Si l'on indique uniquement un nom d'utilisateur (ou un UID numérique), celui-ci devient le propriétaire des fichiers donnés, mais les groupes de ces fichiers ne sont pas modifiés. Si le nom d'utilisateur est suivi d'un

BTS SNIR	Document ressource
Lycée Jean Rostand Villepinte	Gestion des fichiers et des utilisateurs sous Debian

deux-points, et d'un nom de groupe (ou d'un GID numérique), les fichiers changent également de groupe.

Options les plus courantes :

- v Affiche des informations sur les modifications apportées ou non aux fichiers.
- R Effectue les modifications de droits de façon récursive.

Exemple : le propriétaire est « bmaheux » et le groupe propriétaire « professeur »

```
chown bmaheux:professeur fichier
```

II.15 Modification des droits

La commande est **chmod**.

Syntaxe :

```
chmod [options] mode fichiers
```

Description :

Cette commande modifie les droits d'accès des fichiers spécifiés. Les droits d'accès (mode) sont codés par une représentation symbolique ou par un nombre octal. La représentation symbolique utilise le format suivant :

[ugoa] [[+|=][rwxXstugo]]

La combinaison des lettres `ugoa` contrôle la catégorie d'utilisateur pour laquelle on modifie les droits d'accès (utilisateur du fichier (u), utilisateurs du même groupe que le fichier (g), utilisateurs n'appartenant pas au groupe du fichier (o), ou tous les utilisateurs (a)). La dernière catégorie (a) est utilisée par défaut.

L'opérateur `+` ajoute un accès, l'opérateur `-` supprime un accès, et l'opérateur `=` assigne un accès.

Les lettres `rwxXstugo` définissent le type d'accès. Voici les plus usitées :

- x r lecture,
- x w écriture,
- x x exécution,
- x s définit l'ID du propriétaire ou du groupe,
- x u droits en cours du propriétaire,
- x g droits en cours des membres du groupe propriétaire,
- x o droits en cours des autres utilisateurs.

Vous pouvez également utiliser la représentation octale. Cette représentation utilise une combinaison de 3 chiffres en octal (de 0 à 7). Le premier chiffre désigne les droits de l'utilisateur, le second ceux du groupe, et le troisième ceux des autres utilisateurs. Pour chacun des droits, on effectue la somme des valeurs octales requises :

- 4 Lecture
- 2 Écriture
- 1 Exécution

Options les plus courantes :

- v Affiche des informations sur les modifications apportées ou non aux fichiers.
- R Effectue les modifications de droits de façon récursive.

Exemples :

- x Tous les droits pour le propriétaire et lecture/exécution pour les autres :

```
chmod 755 fichier
```

- x Les droits d'exécution pour tous :

```
chmod +x fichier
```

BTS SNIR	Document ressource
Lycée Jean Rostand Villepinte	Gestion des fichiers et des utilisateurs sous Debian

II.16 Modification du groupe propriétaire

La commande pour ne modifier que le groupe propriétaire est **chgrp**.

Syntaxe :

```
chgrp [options] groupe fichiers
```

Description :

chgrp change l'appartenance de chacun des fichiers indiqués pour qu'ils deviennent propriétés du groupe indiqué.

Options les plus courantes :

- v Affiche des informations sur les modifications apportées ou non aux fichiers.
- R Effectue les modifications de droits de façon récursive.

Exemple : tous les fichiers, répertoires et sous-répertoires du répertoire /home/bmaheux appartiendront au groupe profs

```
chgrp -R profs /home/bmaheux/
```

III Les groupes

Comme nous avons vu les utilisateurs (comptes) sont regroupés dans des groupes ayant des points communs. Il est plus judicieux avant de créer des utilisateurs, de créer les groupes si ils ne font pas partie d'un groupe déjà existant.

Ces groupes sont très pratiques pour donner les mêmes droits à tous tels que tous les utilisateurs d'un même service dans une entreprise. Par exemple, pour une école, on peut créer un groupe pour les élèves et un pour les professeurs par exemple.

III.1 Ajout d'un groupe

Syntaxe :

```
groupadd [-g GID [-o]] [-f] [-K CLÉ=VALEUR] groupe
```

Description :

La commande **groupadd** crée un nouveau compte de groupe en utilisant les valeurs spécifiées sur la ligne de commande et les valeurs par défaut du système. Le nouveau groupe sera inséré dans les fichiers du système selon les besoins.

Options la plus courante :

- g GID Définit l'identifiant du groupe (« group ID » ou GID)

Exemple : création du groupe professeur

```
groupadd professeur
```

III.2 Suppression d'un groupe

Syntaxe :

```
groupdel groupe
```

Description :

La commande **groupdel** modifie les fichiers d'administration des comptes du système, en supprimant les entrées qui se réfèrent à groupe. Le groupe indiqué doit exister.

Exemple : suppression du groupe professeurs

```
groupdel professeurs
```

III.3 Modification d'un groupe

Syntaxe :

```
groupmod [options] GROUPE
```

Description :

La commande **groupmod** modifie la définition du groupe spécifié (GROUPE).

BTS SNIR	Document ressource
Lycée Jean Rostand Villepinte	Gestion des fichiers et des utilisateurs sous Debian

Options les plus courantes :

- g **GID** Définit l'identifiant du groupe (« group ID » ou GID)
- n **NOUVEAU_NOM** Définit le nom du groupe

Exemple : le groupe professeur s'appellera professeurs

```
groupmod -n professeurs professeur
```

IV Les comptes utilisateur

Pour créer un nouveau compte utilisateur, il faut exécuter plusieurs commandes :

- x **useradd** : pour créer le compte ;
- x **chfn** : pour donner les informations sur l'utilisateur ;
- x **passwd** : pour définir le mot de passe de cet utilisateur.

IV.1 Ajout d'un utilisateur

La commande **useradd** permet, de créer un nouvel utilisateur.

Syntaxe :

useradd

```
[-c commentaire] [-d rép_perso]
[-e date_expiration] [-f inactive_time]
[-g groupe_initial] [-G groupe[,...]]
[-m [-k rép_squelette] | -M] [-p mot_de_passe]
[-s shell] [-u uid [-o]] [-n] [-r] login
```

Description :

Quand elle est invoquée sans l'option **-D**, la commande **useradd** crée un nouveau compte utilisateur qui utilise les valeurs spécifiées sur la ligne de commandes, et les valeurs par défaut du système. Le nouveau compte utilisateur sera inséré dans les fichiers du système quand c'est nécessaire, le répertoire personnel sera créé, et les fichiers initiaux copiés, cela dépendant des options présentes en ligne de commandes. La version fournie avec Red Hat Linux crée un groupe pour chaque utilisateur ajouté au système, à moins que l'option **-n** ne soit fournie.

Options les plus courantes :

- g **groupe** Définit le nom du groupe ou le numéro du groupe de connexion de l'utilisateur.
- m Le répertoire personnel de l'utilisateur sera créé s'il n'existe pas déjà.
- s **shell** Définit le nom du shell de connexion de l'utilisateur.

Exemple : création du compte « bmaheux » du groupe « professeur » avec création du répertoire « /home/bmaheux » et utilisation du shell « /bin/bash »

```
sudo useradd -g professeur -m -s /bin/bash bmaheux
```

IV.2 Configuration d'un utilisateur

Pour modifier les informations sur un utilisateur, il faut utiliser la commande **chfn**.

Syntaxe :

```
chfn [-f nom_complet] [-r no_bureau] [-w tel_bureau] [-h tel_perso]
[-o autre] [utilisateur]
```

Description :

chfn modifie le nom complet d'un utilisateur, son numéro de bureau, son numéro de téléphone professionnel, et son numéro de téléphone personnel. Un utilisateur normal ne peut modifier que les informations associées à son propre compte, avec les restrictions spécifiées dans */etc/login.defs*. (Par défaut, les utilisateurs ne peuvent pas modifier leur nom complet). Le superutilisateur peut modifier n'importe quel champ pour n'importe quel compte.

Si aucune option n'est sélectionnée, **chfn** opère de manière interactive, demandant à l'utilisateur d'entrer les valeurs actuelles de chacun des champs. Entrer une nouvelle valeur pour la modifier, ou

BTS SNIR	Document ressource
Lycée Jean Rostand Villepinte	Gestion des fichiers et des utilisateurs sous Debian

de laisser une ligne blanche pour conserver la valeur actuelle. La valeur actuelle est indiquée entre crochets ([]). En l'absence d'option, **chfn** opère sur l'utilisateur actuel.

Exemple : Changer le nom complet par « Barbara Maheux »

```
chfn -f "Barbara Maheux" bmaheux
```

IV.3 Changement du mot de passe d'un utilisateur

Pour changer le mot de passe, la commande **passwd** doit être utilisée.

Syntaxe :

```
passwd [options] [LOGIN]
```

Description :

La commande **passwd** modifie les mots de passe des comptes des utilisateurs. Un utilisateur normal peut uniquement modifier le mot de passe de son compte (le plus souvent), quand le Super-utilisateur a le pouvoir de changer tous les mots de passe des comptes. **passwd** permet aussi d'associer une période de validité aux comptes et mots de passe, et de les modifier.

Exemple : Modification du mot de passe de maheux

```
root@debian-1:~# passwd maheux
Entrez le nouveau mot de passe UNIX :
Retapez le nouveau mot de passe UNIX :
passwd : le mot de passe a été mis à jour avec succès
```

IV.4 Changement de l'utilisateur courant

Pour passer d'un utilisateur à un autre, il faut entrer la commande **su** suivie du nom de l'utilisateur.

Syntaxe :

```
su [-flmp] [-c commande] [-s shell] [--login] [--fast]
  [--preserve-environment]      [--command=commande]
  [--shell=shell] [-] [--help] [--version] [utilisateur [arg...]]
```

Description :

su permet à un utilisateur de se transformer temporairement en un autre utilisateur.

Si aucun nom d'utilisateur n'est mentionné, le nom **root**, le Super-Utilisateur, est utilisé par défaut.

Un shell est exécuté avec les UID, GID effectifs et réels, ainsi que les groupes supplémentaires de l'utilisateur indiqué.

Le shell exécuté est celui mentionné dans l'enregistrement correspondant à l'utilisateur dans le fichier **/etc/passwd**. Si aucun shell n'y est indiqué, **/bin/sh** est exécuté par défaut.

Si l'utilisateur dispose d'un mot de passe, **su** le réclame, à moins qu'il n'ait été invoqué avec un UID réel valant 0 (celui du Super-Utilisateur).

Par défaut, **su** ne change pas de répertoire.

IV.5 Commande **userdel**

Comme il est possible de créer un utilisateur, il est possible de le supprimer grâce à la commande **userdel**.

Syntaxe :

```
userdel [-r] login
```

Description :

La commande **userdel** modifie les fichiers des comptes du système, en supprimant toutes les entrées qui se réfèrent à **login**. L'utilisateur nommé doit exister.

Option :

-r Les fichiers présents dans le répertoire personnel de l'utilisateur seront supprimés en même temps que le répertoire lui-même.

BTS SNIR	Document ressource
Lycée Jean Rostand Villepinte	Gestion des fichiers et des utilisateurs sous Debian

IV.6 Modification des paramètres d'un utilisateur

La commande **usermod** comme son nom l'indique permet de modifier les paramètres d'un utilisateur tel que le shell ou le répertoire personnel.

Syntaxe :

```
usermod
[-c commentaire] [-d rép_perso [-m]]
[-e date_expiration] [-f inactivité]
[-g groupe_initial] [-G groupe[,...]]
[-l nom_connexion] [-p mot_de_passe]
[-s shell] [-u uid [-o]] [-L|-U] login
```

Description :

La commande **usermod** modifie les fichiers des comptes du système pour refléter les modifications qui ont été spécifiées sur la ligne de commandes.

Options les plus courantes :

- d rép_perso Définit le nouveau répertoire de connexion de l'utilisateur.
- e date_expiration Définit la date à laquelle le compte utilisateur sera désactivé. La date est spécifiée dans le format AAAA-MM-JJ.
- g groupe Définit le nom ou le numéro du groupe de connexion de l'utilisateur.
- G groupe[,...] Définit une liste de groupes supplémentaires auxquels fait également partie l'utilisateur.
- l login Définit le nom de l'utilisateur .
- p mot_de_passe Définit le mot de passe chiffré.
- s shell Définit le nom du nouveau shell de connexion de l'utilisateur.

Exemple : l'identifiant de compte de maheux est bmaheux et son dossier personnel se trouve à l'emplacement /home/bmaheux/.

```
usermod --login bmaheux --home /home/bmaheux --move-home maheux
```

Exemple : le compte bmaheux est ajouté au groupe (secondaire) administrateurs

```
usermod -aG administrateurs bmaheux
```

IV.7 Gestion des groupes d'un utilisateur

La commande **gpasswd** permet de gérer les groupes d'un utilisateur.

Syntaxe :

```
gpasswd -a utilisateur groupe
gpasswd -d utilisateur groupe
```

Options les plus courantes :

- a ajoute un utilisateur à un groupe.
- d supprime un utilisateur d'un groupe.

Exemple : ajout bmaheux au groupe userFTP

```
root@debian0:~# gpasswd -a bmaheux userFTP
Ajout de l'utilisateur bmaheux au groupe userFTP
```

IV.8 Affichage des groupes d'un utilisateur

La commande **groups** permet de lister tous les groupes (primaire et secondaires) d'un utilisateur.

Syntaxe :

```
groups [utilisateur...]
```

Exemple : Affichage des groupes dont fait partie bmaheux

BTS SNIR	Document ressource
Lycée Jean Rostand Villepinte	Gestion des fichiers et des utilisateurs sous Debian

```
root@debian0:~# groups bmaheux
bmaheux : professeur userFTP
```