

# Utilisateurs et droits sous Linux

*Attribution - Partage dans les Mêmes Conditions :*  
<http://creativecommons.org/licenses/by-sa/3.0/fr/>

# Table des matières

|   |           |
|---|-----------|
| <b>I - Contexte</b>                                 | <b>3</b>  |
| <b>II - Les utilisateurs et les groupes</b>         | <b>4</b>  |
| <b>III - Exercice : Appliquer la notion</b>         | <b>7</b>  |
| <b>IV - L'utilisateur root</b>                      | <b>8</b>  |
| <b>V - Exercice : Appliquer la notion</b>           | <b>10</b> |
| <b>VI - Configuration des utilisateurs</b>          | <b>11</b> |
| <b>VII - Exercice : Appliquer la notion</b>         | <b>14</b> |
| <b>VIII - Configuration des groupes</b>             | <b>15</b> |
| <b>IX - Exercice : Appliquer la notion</b>          | <b>17</b> |
| <b>X - Affichage des droits sous Linux (ls -al)</b> | <b>18</b> |
| <b>XI - Exercice : Appliquer la notion</b>          | <b>20</b> |
| <b>XII - Gestion des droits sous Linux</b>          | <b>21</b> |
| <b>XIII - Exercice : Appliquer la notion</b>        | <b>23</b> |
| <b>XIV - Quiz</b>                                   | <b>24</b> |
| <b>Solutions des exercices</b>                      | <b>27</b> |
| <b>Contenus annexes</b>                             | <b>33</b> |

# I Contexte

**Durée** : 2h

**Environnement de travail** : Linux en ligne de commande

**Pré-requis** : Savoir utiliser la ligne de commande

Linux est un système multi-utilisateurs, ce qui signifie que plusieurs utilisateurs différents peuvent utiliser le système en même temps. Ces utilisateurs peuvent être des utilisateurs humains, mais aussi des utilisateurs systèmes qui sont en charge de faire exécuter différents logiciels en arrière-plan. Cette composante multi-utilisateurs est très importante, et elle nécessite une organisation et une gestion des droits simple mais stricte pour fonctionner correctement.

Nous allons voir dans ce cours comment sont organisés les utilisateurs, comment les configurer, mais aussi la gestion des droits et la notion de super-utilisateur.

## II Les utilisateurs et les groupes

### Objectifs

- Comprendre ce qu'est un utilisateur Linux
- Comprendre ce qu'est un groupe
- Savoir interpréter des UID/GID

### Les utilisateurs, au cœur du système

Sous Linux, tout les fichiers et dossiers appartiennent à un utilisateur, et tout les programmes qui s'exécutent sont exécutés par un utilisateur donné. En fait, plus ou moins tout est lié à un ou plusieurs utilisateurs. On ne parle pas forcément d'utilisateurs humains, mais d'utilisateurs au sens informatique. Beaucoup de logiciels en arrière-plan s'exécute avec un utilisateur défini uniquement pour cet usage. Pour illustrer ça, on peut afficher le fichier `/etc/passwd` qui contient de nombreuses informations, en particulier les utilisateurs qui existent. Par exemple sur ma machine j'ai ceci :

```
1 $ cat /etc/passwd
2 root:x:0:0:root:/root:/bin/bash
3 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
4 bin:x:2:2:bin:/bin:/usr/sbin/nologin
5 sys:x:3:3:sys:/dev:/usr/sbin/nologin
6 sync:x:4:65534:sync:/bin:/bin/sync
7 games:x:5:60:games:/usr/games:/usr/sbin/nologin
8 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
9 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
10 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
11 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
12 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
13 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
14 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
15 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
16 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
17 irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
18 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
19 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
20 systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
21 systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/false
22 systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
23 messagebus:x:104:111:./var/run/dbus:/bin/false
24 avahi:x:105:112:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
25 uidd:x:106:114:./run/uidd:/bin/false
26 Debian-exim:x:107:115:./var/spool/exim4:/bin/false
27 statd:x:108:65534:./var/lib/nfs:/bin/false
28 avahi-autoipd:x:109:118:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
29 colord:x:110:120:colord colour management daemon,,,:/var/lib/colord:/bin/false
30 dnsmasq:x:111:65534:dnsmasq,,,:/var/lib/misc:/bin/false
31 sshd:x:113:65534:./var/run/sshd:/usr/sbin/nologin
32 pulse:x:114:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false
33 rtkit:x:115:124:RealtimeKit,,,:/proc:/bin/false
34 saned:x:116:125:./var/lib/saned:/bin/false
35 usbmux:x:117:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
36 hplip:x:119:7:HPLIP system user,,,:/var/run/hplip:/bin/false
37 kyane:x:1000:1000:Kyâne,,,:/home/kyane:/usr/bin/zsh
38 geoclue:x:120:127:./var/lib/geoclue:/bin/false
```

```

39 _apt:x:121:65534:./nonexistent:/bin/false
40 Debian-minetest:x:122:60:Minetest dedicated server,,,:/var/games/minetest-
   server:/bin/false
41 vde2-net:x:123:132:./var/run/vde2:/bin/false
42 radvd:x:124:65534:./var/run/radvd:/bin/false
43 snort:x:125:135:Snort IDS:/var/log/snort:/bin/false
44 ntp:x:103:106:./nonexistent:/usr/sbin/nologin
45 systemd-coredump:x:997:997:systemd Core Dumper:./sbin/nologin
46 debian-tor:x:126:137:./var/lib/tor:/bin/false
47 _rpc:x:127:65534:./run/rpcbind:/usr/sbin/nologin
48 tcpdump:x:112:126:./nonexistent:/usr/sbin/nologin

```

Il n'est pas nécessaire de tout comprendre pour le moment, il faut simplement savoir que le texte indiqué en début de chaque ligne, avant les ":", est le nom d'un utilisateur. On voit qu'il y en a vraiment beaucoup, et en réalité il n'y en a qu'une seul qui correspond à un être humain, c'est kyane, ligne 37. Tout le reste sont des utilisateurs dits "système", dans le sens où ils servent plus au système d'exploitation qu'à un utilisateur. Par exemple le dernier, tcpdump, est utilisé par le programme tcpdump (un outil de capture du flux réseau) pour réaliser les opérations qu'il a besoin de faire.

## Les groupes

On peut créer autant d'utilisateurs que l'on veut sur Linux, et les répartir dans des groupes. Un groupe est, comme son nom l'indique, un regroupement de plusieurs utilisateurs, ce qui permettra ensuite d'appliquer des règles à tout les utilisateurs d'un même groupe d'un coup. Par défaut chaque utilisateur se trouve dans un groupe qui porte le même nom (l'utilisateur kyane est par défaut dans le groupe kyane), c'est ce que l'on appelle le groupe primaire. Mais un utilisateur peut-être ajouté à plusieurs autres groupes par la suite, par exemple le même utilisateur kyane se trouve aussi dans un groupe bluetooth, qui me permet d'utiliser le Bluetooth de mon ordinateur.

### UID et GID

Az Définition

Les UID et GID sont les identifiants des utilisateurs et groupes. En effet le système d'exploitation ne manipule pas directement des noms textuels, mais des identifiants numériques. L'UID, pour *User ID*, est l'identifiant qui correspond à un utilisateur donné. Dans l'exemple de l'utilisateur kyane plus haut, son UID est 1000. Le GID pour *Group ID* est la même chose mais pour un groupe.

### La commande id

Méthode

La commande `id` permet d'obtenir très simplement une information sur l'utilisateur avec lequel on est connecté, ainsi que les groupes associés.

```

1 $ id
2 uid=1000(kyane) gid=1000(kyane)
   groupes=1000(kyane),4(adm),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plu

```

Ici on voit en premier que mon UID est 1000, c'est à dire l'utilisateur kyane, et mon GID est aussi 1000, avec un groupe du même nom (mon groupe primaire). Ensuite on voit la liste des groupes (nom et GID) auquel appartient l'utilisateur.

## **À retenir**

Les utilisateurs et les groupes sont au cœur du fonctionnement de Linux. Ceux-ci possèdent des identifiants numériques, mais on peut manipuler une traduction sous la forme d'un nom textuel, pour les humains.

# III Exercice : Appliquer la notion

## Question 1

[solution n°1 p. 27]

Je viens d'ouvrir un terminal sur une machine, mais je ne sais pas quel est l'utilisateur avec lequel je suis connecté, ni les groupes auxquels il appartient. Avec quelle commande est-il possible d'obtenir l'information ?

## Interprétation

Le retour de la commande précédente est le suivant :

```
1 uid=700(kyane) gid=700(kyane)
  groups=700(kyane),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),109(n
```

## Question 2

[solution n°2 p. 27]

Quel est le nom et l'UID de l'utilisateur ?

## Question 3

[solution n°3 p. 27]

Quel est le GID du groupe nommé docker ?

# IV L'utilisateur root

## Objectifs

- Savoir ce qu'est le super-utilisateur
- Savoir utiliser le compte root.

## Un super-utilisateur pour les gouverner tous

Les utilisateurs et groupes peuvent se voir autoriser différents droits sur le système. Ce sera abordé un peu plus loin, mais par exemple l'utilisateur par défaut que l'on a créé à l'installation ne pourra pas forcément lire tout les fichiers d'un autre utilisateur (et cela semble normal). On peut donc se demander « *qui créé les utilisateurs ?* », « *qui peut donner des droits ?* ».

La réponse est simple : le super-utilisateur. C'est un utilisateur un peu particulier car il a absolument tout les pouvoirs sur la machine. Sous Linux cet utilisateur s'appelle `root`, qui signifie *racine*.

### Super-utilisateur root

💡 Fondamental

L'utilisateur `root` est le nom du super-utilisateur sous Linux. Il est le seul à avoir un UID spécial : 0. Il a la possibilité de réaliser toutes les actions et modifications possibles sur le système.

👁 Exemple

Sur votre installation Linux, essayez de lister les fichiers dans le dossier `/root` à l'aide de votre utilisateur classique.

```
1 $ ls /root
2 ls: impossible d'ouvrir le répertoire '/root': Permission non accordée
```

Nous détaillerons les permissions par la suite, mais votre utilisateur ne peut pas réaliser cette action. Seul l'utilisateur `root` peut le faire.

### Élévation de privilèges

🔧 Méthode

Puisque l'on ne peut pas faire certaines actions en tant qu'utilisateur normal, il existe une technique pour obtenir des privilèges de super-utilisateur, c'est à dire pouvoir se connecter avec le compte `root`. Pour cela, on utilise la commande `sudo`. Elle permet de lancer une commande en tant que super utilisateur.

```
1 $ sudo ls /root
2 [sudo] Mot de passe de kyane :
3 fichier_secret
```

En préfixant notre commande précédente par `sudo` on indique au système que l'on souhaite lancer la commande en tant que `root`. Le mot de passe de notre utilisateur est demandé, puis la commande est exécuté. C'est ce que l'on appelle une élévation de privilèges.

Cependant, tout le monde n'a pas le pouvoir de lancer des commandes en utilisant `sudo`, ce serait beaucoup trop facile, n'importe quel utilisateur pourrait obtenir tous les droits sur la machine. Cela n'est possible que pour un groupe d'utilisateurs bien précis : ceux du groupe `sudo`. Avec la commande `id` vous pourrez constater que votre utilisateur est bien dans ce groupe, ce qui est le cas pour l'utilisateur créé lors de l'installation.

### Une console root

 Méthode

Parfois on souhaite réaliser plusieurs opérations en tant que super-utilisateur, et il n'est pas très pratique de voir écrire `sudo` au début de toutes les commandes. Pour cela on peut ouvrir une console en tant que `root`, pour que toutes nos commandes puissent avoir le privilège super-utilisateur.

```
1 kyane@europa:~$ sudo -i
2 root@europa:~#
```

Après avoir entré la commande `sudo -i`, on remarque que le prompt est modifié : on est l'utilisateur `root` et le `#` nous confirme que c'est bien un super-utilisateur. On peut désormais lancer plusieurs commandes en mode privilégié.

### With great power comes great responsibility

 Attention

Lorsque l'on exécute des commandes en tant que `root`, on peut **tout faire** sur la machine. Et quand l'on ne fait pas attention à ce que l'on fait, on a le pouvoir de tout casser. Il n'est pas recommandé de lancer toutes les commandes en `root` "juste pour être sûr", mais plutôt de n'utiliser `sudo` que lorsque nécessaire, et après une réflexion sur les actions qui sont faites.

## À retenir

Le super-utilisateur sur Linux s'appelle `root` et c'est celui qui a tous les pouvoirs sur la machine. Tous les autres utilisateurs ont des droits limités. Cependant il est possible d'utiliser la commande `sudo` pour élever ses privilèges lorsque nécessaire.

## V Exercice : Appliquer la notion

Le fichier `/etc/shadow` sur un environnement Linux contient des informations sensibles, entre autre les mots de passe (chiffrés) de certains utilisateurs.

### Question 1

[solution n°4 p. 27]

Affichez le contenu de ce fichier dans votre terminal.

#### Indice :

Normalement vous n'avez pas les droits pour le faire avec votre utilisateur classique.

### Question 2

[solution n°5 p. 27]

Nous savons que l'UID de l'utilisateur `root` est 0, mais comment le vérifier ?

# VI Configuration des utilisateurs

## Objectifs

- Savoir trouver les informations de configuration des utilisateurs
- Savoir créer et supprimer des utilisateurs

## Le fichier `/etc/passwd`

Un peu plus tôt nous avons ouvert le fichier `/etc/passwd` qui contient la liste des utilisateurs de la machine. Reprenons une ligne de ce fichier pour voir toutes les informations qu'il contient.

```
1 kyane:x:1000:1000:Kyâne:/home/kyane:/bin/bash
```

Chaque information est séparée par un caractère ":". Dans l'ordre nous avons :

- le nom d'utilisateur `kyane` ;
- le caractère "x" qui indique que l'utilisateur a un mot de passe (stockée dans `/etc/shadow`), en effet il peut exister des utilisateurs sans mot de passe ;
- l'UID, ici `1000` ;
- le GID, ici `1000` ;
- une description du compte utilisateur, ici `Kyâne` ;
- le dossier de l'utilisateur, que l'on appelle son `home` ;
- le *shell* utilisé par l'utilisateur, c'est à dire le programme qui interprète ses commandes (il en existe en fait plusieurs). Par défaut c'est généralement `/bin/bash`.

### Manipuler les utilisateurs

 Attention

Maintenant que l'on sait où sont configurés les utilisateurs, on est tenté de modifier le fichier `/etc/passwd` pour créer ou supprimer des utilisateurs. En réalité ce n'est pas recommandé, on préfère laisser le système s'occuper de cela pour nous et plutôt utiliser des commandes adaptées.

### Créer un utilisateur

 Méthode

Pour créer un utilisateur, on utilise tout simplement la commande `adduser`. Attention, c'est une commande qui nécessite les privilèges `root`. Par exemple voici la commande et le retour si je veux créer un utilisateur `sarah`.

```
1 sudo adduser sarah
```

```
kyane@europa:~$ sudo adduser sarah
Ajout de l'utilisateur « sarah » ...
Ajout du nouveau groupe « sarah » (1001) ...
Ajout du nouvel utilisateur « sarah » (1001) avec le groupe « sarah » ...
Création du répertoire personnel « /home/sarah »...
Copie des fichiers depuis « /etc/skel »...
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd: password updated successfully
Changing the user information for sarah
Enter the new value, or press ENTER for the default
  Full Name []: Sarah
  Room Number []:
  Work Phone []: 0123456789
  Home Phone []:
  Other []: Compte invite
Cette information est-elle correcte ? [0/n]0
kyane@europa:~$
```

C'est une commande interactive, qui demande de rentrer différentes informations sur l'utilisateur. Hormis le mot de passe qui est obligatoire, les autres informations sont celles qui se trouveront dans la description de l'utilisateur, et qui sont donc facultatives. Pour le reste, tout s'est fait automatiquement : la création de l'utilisateur, du groupe associé, du dossier de l'utilisateur, etc.

La commande a créé la ligne suivante dans le fichier `/etc/passwd`.

```
1 sarah:x:1001:1001:Sarah,,0123456789,,Compte invite:/home/sarah:/bin/bash
```

## Supprimer un utilisateur

 Méthode

Si l'on souhaite supprimer l'utilisateur on utilise la commande suivante.

```
1 kyane@europa:~$ sudo deluser --remove-home sarah
2 Recherche des fichiers à sauvegarder ou à supprimer...
3 Suppression des fichiers...
4 Suppression de l'utilisateur « sarah »...
5 Attention ! Le groupe « sarah » ne contient plus aucun membre.
6 Fait.
7 kyane@europa:~$
```

L'option `--remove-home` n'est pas obligatoire et permet, comme son nom l'indique, de supprimer aussi le répertoire de l'utilisateur en question. Ce n'est pas le cas par défaut, par mesure de sécurité.

## Changer d'utilisateur

 Complément

Pour vous connecter avec votre nouvel utilisateur, vous pouvez bien entendu vous connecter avec son nom et mot de passe au démarrage de la session, mais vous pouvez aussi changer d'utilisateur directement dans votre terminal.

```
1 kyane@europa:~$ su sarah
2 Mot de passe :
3 sarah@europa:/home/kyane$
```

La commande `su` (pour *Substitute User*) permet de se substituer à un autre utilisateur, et donc de se connecter sous cet utilisateur (en utilisant son mot de passe bien entendu). D'ailleurs si vous avez les privilèges super-administrateur, vous pouvez même utiliser `su` sans avoir besoin du mot de passe de l'utilisateur, par exemple en faisant `sudo su sarah`. Le super-utilisateur est donc super-puissant (désolé) comme il peut se faire passer pour n'importe qui !

## À retenir

Les utilisateurs sont configurés dans le fichier `/etc/passwd`, mais il est possible et préférable d'utiliser les commandes `adduser` et `deluser` pour les manipuler.

# VII Exercice : Appliquer la notion

## Créer des utilisateurs

On souhaite créer 2 utilisateurs sur la machine, alice et bob, mais avec certaines contraintes :

- alice aura pour UID 1234
- le dossier d'utilisateur de bob sera `/usr/etc/bob`

Pour le reste, la description des utilisateurs est libre.

### Question 1

[solution n°6 p. 27]

Créez les 2 utilisateurs sur votre machine.

#### Indice :

Il va sans doute falloir lire le manuel de la commande `adduser`.

## Supprimer les utilisateurs

Ces utilisateurs ne nous serviront plus par la suite, on peut donc simplement les supprimer.

### Question 2

[solution n°7 p. 27]

Supprimez les deux utilisateurs en prenant soin à bien retirer leurs dossiers respectifs.

# VIII Configuration des groupes

## Objectifs

- Savoir trouver les informations de configuration des groupes
- Savoir créer et supprimer des groupes
- Savoir assigner des utilisateurs à des groupes

## Le fichier `/etc/group`

Concernant les groupes, c'est le fichier `/etc/group` qui est en charge de gérer leur configuration. Si on observe son contenu on constate qu'il a un format proche de `/etc/passwd`.

```
1 $ cat /etc/group
2 root:x:0:
3 daemon:x:1:
4 bin:x:2:
5 sys:x:3:
6 adm:x:4:kyane
7 tty:x:5:
8 disk:x:6:
9 lp:x:7:
10 mail:x:8:
11 news:x:9:
12 uucp:x:10:
13 man:x:12:
14 proxy:x:13:
15 kmem:x:15:
16 dialout:x:20:
17 fax:x:21:
18 voice:x:22:
19 cdrom:x:24:kyane
20 floppy:x:25:kyane
21 tape:x:26:
22 sudo:x:27:kyane
23 audio:x:29:pulse,kyane
```

Il contient en fait la liste des groupes de la machine, avec un format proche (ici l'aperçu qui est donné est volontairement tronqué, il y a sans doute plus de lignes dans votre fichier). Isolons la dernière ligne pour l'étudier.

```
1 audio:x:29:pulse,kyane
```

Les informations sont, ici aussi, séparées par des ":". Dans l'ordre nous avons :

- le nom du groupe `audio` ;
- un caractère "x" qui n'a, en fait, pas d'importance dans ce fichier ;
- le GID du groupe, ici 29 ;
- la liste, séparée par des virgules, des utilisateurs faisant partie du groupe.

On peut donc lire la ligne suivante de cette manière : le groupe `audio` a pour GID 29 et les utilisateurs `pulse` et `kyane` sont présents dans ce groupe.

## Créer un groupe

[🔗 Méthode](#)

Mais si l'on souhaite grouper plusieurs utilisateurs dans un groupe, par exemple pour gérer leurs droits, on a besoin de créer un groupe. Pour cela un utilisateur, on utilise tout simplement la commande `addgroup`.

```
1 kyane@europa:~$ sudo addgroup enfants
2 Ajout du groupe « enfants » (GID 1001)...
3 Fait.
```

On voit que la commande a créé un groupe nommé `enfants` sur la machine. Le fichier `/etc/group` a été modifié en conséquence.

```
1 enfants:x:1001:
```

Pour le moment notre groupe est complètement vide. Pour ajouter des utilisateurs à un groupe, on peut utiliser la commande `usermod`. C'est une commande qui permet de modifier de nombreux attributs d'un utilisateur, ou même de le renommer. Nous n'allons pas nous intéresser à toutes les possibilités offertes par cette commande, mais uniquement à l'utilisation pour modifier les groupes d'un utilisateur.

## Modifier les groupes d'un utilisateur

[🔗 Méthode](#)

À l'aide des options `-aG` il est possible d'ajouter un groupe à un utilisateur.

```
1 sudo usermod -aG enfants roxane
```

L'utilisateur `roxane` fait désormais partie du groupe `enfants`. L'option `G` permet d'indiquer qu'on modifie les groupes de l'utilisateur, et a permis d'*append* (ajouter) un groupe. On peut le vérifier dans le fichier `/etc/group`.

```
1 enfants:x:1001:roxane
2 roxane:x:1002:roxane
```

## Supprimer un groupe

[🔗 Méthode](#)

De manière assez évidente, la commande `delgroup` permet de supprimer un groupe.

```
1 kyane@europa:~$ sudo delgroup enfants
```

## À retenir

Le fichier `/etc/group` contient la configuration des groupes sur la machine, et les commandes `addgroup` et `delgroup` permettent de les manipuler. On utilise ensuite `usermod`, en particulier avec ses options `-aG`, pour modifier les groupes auxquels appartient un utilisateur.

## IX Exercice : Appliquer la notion

On souhaite créer 3 utilisateurs sur notre machine :

- joe
- paul
- mike

On souhaite aussi avoir 2 groupes :

- administrateurs dont feront partie joe et paul
- utilisateurs dont feront partie joe et mike

On fera en sorte que les 3 utilisateurs conservent leur groupe primaire d'origine.

### Question

[solution n°8 p. 28]

Créer les 2 utilisateurs et groupes sur la machine

# X Affichage des droits sous Linux (ls -al)

 Rappel

Commandes de bases sous Linux : *cd, ls, mkdir, rm, find, cat, nano...* (cf. p.33)

Sous Linux, chaque dossier et chaque fichier :

- appartient à un utilisateur **et** à un groupe ;
- possède des droits.

Ces droits se divisent en trois :

1. Les droits pour l'utilisateur propriétaire
2. Les droits pour les utilisateurs membres du groupe propriétaire
3. Les droits pour tous les autres utilisateurs

## ls -al

 Fondamental

La commande `ls -al` affiche tous les fichiers et dossiers d'un répertoire, avec leurs propriétaires et leurs droits.

```
1 drwxr-x--- 3 stc friends 4096 mars 31 19:11 .dir
```

- Le premier caractère indique que l'élément est un dossier (*d*) ou un fichier (*-*).
- Les trois caractères suivants indiquent les droits de l'utilisateur propriétaire (ici *stc* a les droits *rw**x* : lire, écrire, parcourir)
- Les trois caractères suivants indiquent les droits des utilisateur membres du groupe propriétaire (ici les membres de *friends* ont les droits *r-x* : lire et parcourir, mais pas écrire)
- Les trois caractères suivants indiquent les droits des autres utilisateurs (ici *---* : aucun droit)
- Le nombre suivant indique le nombre de sous-dossiers dans le dossier listé (ici *.dir* a 3 sous-dossiers, dont *.* et *..* donc un seul vrai sous-dossier)
- La chaîne suivante indique l'utilisateur propriétaire (ici l'utilisateur *stc*)
- La chaîne suivante indique le groupe propriétaire (ici le groupe *friends*)
- Le nombre suivant indique la taille de l'élément en octets (ici 4096 octets)
- La chaîne suivante indique la date et l'heure de dernière modification du fichier (ici le 31 mars à 19h11)
- La dernière chaîne est le nom du fichier (ici *.dir*)

 Exemple

```
1 -rw----- 1 root root 2048 mars 31 19:11 .config
2 drwx----- 2 stc stc 4096 avril 3 20:25 firefox
3 -rw-rw-r-- 1 stc www 31691 avril 2 10:11 test.jpg
```

- Le fichier `.config` appartient à l'utilisateur `root` et au groupe `root` (il y a souvent un utilisateur et un groupe de même nom) ; l'utilisateur `root` peut le lire et le modifier, personne d'autre ne peut y accéder.
- Le dossier `firefox` appartient à l'utilisateur `stc` et au groupe `stc` ; l'utilisateur `stc` peut y accéder, le lire et le modifier, personne d'autre ne peut y accéder.
- Le fichier `test.jpg` appartient à l'utilisateur `stc` et au groupe `www` ; l'utilisateur `stc` ainsi que les membres du groupe `www` peuvent le lire et le modifier, les autres utilisateurs peuvent seulement le lire.

### Les droits rwx

💡 Fondamental

- **r** (*read*) : lire le contenu d'un fichier (exécuter `more` par exemple) ou d'un dossier (`ls`)
- **w** (*write*) : écrire un fichier (`nano`), écrire dans un dossier (`touch`)
- **x** (*execute*) : exécuter un fichier, entrer dans un dossier (`cd`)

⊕ Complément

<http://www.linuxcertif.com/doc/keyword/ls/>

<http://www.linux-france.org/article/sys/fichiers/>

# XI Exercice : Appliquer la notion

On se demande quels sont les droits par défaut d'un fichier que l'on vient de créer sur notre machine.

## Question 1

[solution n°9 p. 28]

Comment obtenir cette information ?

## Question 2

[solution n°10 p. 28]

Quels sont les droits par défaut pour le propriétaire ? Pour le groupe ? Pour les autres utilisateurs ?

## XII Gestion des droits sous Linux

 Rappel

Affichage des droits sous Linux (cf. p.18)

### chmod

 Fondamental

La commande `chmod` permet de modifier l'attribution des droits d'un fichier ou d'un dossier. Pour cela on ressort d'un côté nos trois types d'utilisateurs : le **user** propriétaire `u`, le **group** propriétaire `g` et les **others** `o`, et d'un autre côté les trois droits **read**, **write** et **execute**.

On donne à `chmod` les modifications que l'on souhaite apporter puis la liste des fichiers/dossiers auquel l'appliquer. Une modification de droit est décrite par `[usertype][action][permissions]` où :

- `usertype` est `u` `g` ou `o` (ou `a` pour all)
- `action` est `+` `-` ou `=` pour ajouter, retirer ou fixer des permissions
- `permissions` est `r` `w` ou `x`

 Exemple

```
1 $ ls -l index.html
2 -rw-rw---- 1 antoine antoine 8578 janv. 22 09:34 index.html
3 $ chmod a=r index.html
4 $ ls -l index.html
5 -r--r--r-- 1 antoine antoine 8578 janv. 22 09:34 index.html
6 $ chmod +x index.html
7 $ ls -l index.html
8 -r-xr-xr-x 1 antoine antoine 8578 janv. 22 09:34 index.html
9 $ chmod o-rx index.html
10 $ ls -l index.html
11 -r-xr-x--- 1 antoine antoine 8578 janv. 22 09:34 index.html
```

Alternativement, les administrateurs système hirsutes préfèrent attribuer 3 nombres de 0 à 7 correspondant aux 3 types d'utilisateurs tel que :

1. `r` vaut 4, `w` vaut 2, `x` vaut 1 (ce sont des puissances de 2 pour ceux qui voudraient une petite explication de comment ça marche n'hésitez pas !);
2. on somme les droits que l'on veut cumuler.

Ainsi on peut régler tous les droits de tous les types d'utilisateurs d'un coup mais c'est moins intuitif.

## 👁 Exemple

- `chmod 640 file` donne les droits **rw** (4+2) à *user*, **r** (4) à *group* et aucun droit (0) à *others*.
- `chmod 775 file` donne les droits **rw**x (4+2+1) à *user* et *group* et **rx** (4+1) à *others*.

**Droits courant pour un fichier (non exécutable)**

## 🔗 Méthode

- `chmod 666 file` (fichier non exécutable public)
- `chmod 644 file` (fichier non exécutable public, modifiable par soi uniquement)
- `chmod 600 file` (fichier privé)
- `chmod 400 file` (fichier privé en lecture seule)
- `chmod 444 file` (fichier public en lecture seule)

**Droits courant pour un dossier (et de son contenu)**

`chmod 777 dir` (dossier public)

`chmod 755 dir` (dossier public, modifiable par soi uniquement)

`chmod 700 dir` (dossier privé)

`chmod 500 dir` (dossier privé en lecture seule)

`chmod 555 dir` (dossier public en lecture seule)

**chmod -R Changer les droits de tout un répertoire**

## ⊕ Complément

`chmod -R XXX dir` affecte les droit XXX à *dir* et à tous les fichiers et sous-dossiers de *dir* (récursivement).

**chown : Changer les propriétaires d'un fichier**

## ⊕ Complément

`chown user:group target` permet de donner la propriété de *target* à l'utilisateur *user* et au groupe *group*.

## XIII Exercice : Appliquer la notion

On reprends les utilisateurs et groupes créés un peu plus tôt, avec les commandes suivantes. On considère que toutes les commandes sont lancées en root.

```
1 adduser paul
2 adduser mike
3 adduser joe
4 addgroup administrateurs
5 addgroup utilisateurs
6 usermod -G paul,administrateurs paul
7 usermod -G joe,administrateurs,utilisateurs joe
8 usermod -G mike,utilisateurs mike
```

### Question

[solution n°11 p. 28]

Dans le dossier /home/joe créez :

- un fichier nommé README que seul joe pour modifier, mais que tout le monde pourra lire
- un dossier nommé admin accessible en lecture et écriture a toutes les personnes du groupe administrateurs, et à personne d'autre
- un fichier dans le dossier admin nommé secret.txt que seul seuls les membres du groupe administrateurs pourront lire et modifier

# XIV Quiz

## Exercice 1 : Quiz - Culture

[solution n°12 p. 29]

### Exercice

Qu'est-ce que le groupe primaire d'un utilisateur Linux ?

**A** Le dernier groupe dans lequel il vient d'être ajouté

**B** Le groupe principal dans lequel il se trouve, généralement attribué à la création de l'utilisateur

**C** Le groupe qu'il partage avec l'utilisateur root

### Exercice

Qu'est-ce qu'un UID ?

**A** Un identifiant unique associé à la machine

**B** Un identifiant associé à un groupe d'utilisateur

**C** Un identifiant associé à un utilisateur

### Exercice

Par défaut il est conseillé d'utiliser tout le temps l'utilisateur root pour administrer la machine.

**A** Vrai

**B** Faux

### Exercice

Dans quel fichier est-il possible de trouver la configuration des utilisateurs de la machine ?

**A** /etc/passwd

**B** /root/passwd

**C** /root/users

**D** /etc/users

## Exercice 6 : Quiz - Méthode

[solution n°13 p. 30]

### Exercice

Pour créer un groupe sur Linux, on préfère :

**A** utiliser la commande `addgroup`

**B** utiliser la commande `groupadd`

**C** modifier le fichier `/etc/group`

### Exercice

Pour lancer une commande avec les privilèges super-utilisateur, quelles sont les différentes possibilités ?

**A** Me connecter avec l'utilisateur `root` au démarrage

**B** Utiliser la commande `sudo -i` pour ouvrir un terminal `root` puis lancer ma commande

**C** Lancer la commande `sudo MACOMMANDE`

**D** Utiliser la commande `root MACOMMANDE`

### Exercice

Si je souhaite donner l'accès à un fichier à plusieurs utilisateurs précis.

**A** Il faut donner des droits en lecture à tout les utilisateurs du système

**B** Ce n'est pas possible

**C** Il faut ajouter les utilisateurs en question dans un groupe et donner les droits de lecture de notre fichier à ce groupe

## Exercice 10 : Quiz - Code

[solution n°14 p. 31]

### Exercice

Quelles sont les affirmations vraies pour un fichier dont `ls -l` affiche les informations suivantes ? `drwxrwxr-x`

**A** Il s'agit d'un dossier

**B** Seul le propriétaire a des droits d'écriture

**C** Tout les utilisateurs peuvent le lire

**D** Il n'est associé à aucun groupe

### Exercice

Quels sont les droits que l'on s'attend à avoir si l'on fait la commande `chmod 640` ?

**A** `rw-rw----`

**B** `rw-rw----`

**C** `r-xr-----`

**D** `rw-r-----`

# Solutions des exercices

## Solution n°1

[exercice p. 7]

On utilise la commande `id`.

## Solution n°2

[exercice p. 7]

L'utilisateur est `kyane`, avec pour UID 700.

## Solution n°3

[exercice p. 7]

Son GID est 998.

## Solution n°4

[exercice p. 10]

On utilise la commande suivante pour afficher le contenu du dossier, en tant qu'utilisateur `root`.

```
1 sudo cat /etc/shadow
```

## Solution n°5

[exercice p. 10]

La commande `id` permet d'obtenir les informations sur un utilisateur. Si on la lance en tant que `root`, on obtient :

```
1 $ sudo id
2 uid=0(root) gid=0(root) groupes=0(root)
```

## Solution n°6

[exercice p. 14]

En lisant le manuel (avec `man adduser`) on se rend compte qu'il est possible d'utiliser une option `--home` pour choisir le chemin du dossier utilisateur et `--uid` pour choisir son UID. On utilise donc les 2 commandes suivantes :

```
1 sudo adduser --home /usr/etc/bob bob
2 sudo adduser --uid 1234 alice
```

On peut vérifier que cela a bien fonctionné en regardant le fichier `/etc/passwd`

```
1 bob:x:1001:1001:,,,:/usr/etc/bob:/bin/bash
2 alice:x:1234:1234:,,,:/home/alice:/bin/bash
```

## Solution n°7

[exercice p. 14]

```
1 sudo deluser --remove-home alice
2 sudo deluser --remove-home bob
```

## Solution n°8

On commence par créer les 3 utilisateurs

```
1 sudo adduser paul
2 sudo adduser mike
3 sudo adduser joe
```

Puis on crée les 2 groupes.

```
1 sudo addgroup administrateurs
2 sudo addgroup utilisateurs
```

Enfin on assigne les bons groupes.

```
1 sudo usermod -aG administrateurs paul
2 sudo usermod -aG administrateurs,utilisateurs joe
3 sudo usermod -aG utilisateurs mike
```

Pour valider, on doit retrouver les lignes suivantes dans les fichiers `/etc/passwd` et `/etc/group`.

```
1 joe:x:1001:1001:,,,:/home/joe:/bin/bash
2 mike:x:1002:1002:,,,:/home/mike:/bin/bash
3 paul:x:1003:1003:,,,:/home/paul:/bin/bash
```

```
1 joe:x:1001:joe
2 mike:x:1002:mike
3 paul:x:1003:paul
4 administrateurs:x:1004:joe,paul
5 utilisateurs:x:1005:joe,mike
```

## Solution n°9

On peut tout simplement créer un fichier avec `touch` et regarder les droits avec `ls -l`.

## Solution n°10

Le fichier créé a les droits suivants :

```
1 kyane@europa:~$ ls -l test.txt
2 -rw-r--r-- 1 kyane kyane 0 4 déc. 15:48 test.txt
3
```

Par défaut le propriétaire est le créateur du fichier, et son groupe est le groupe primaire du créateur.

- Le propriétaire a les droits de lecture et écriture.
- Le groupe a uniquement les droits de lecture.
- Les autres utilisateurs ont aussi les droits de lecture par défaut.

## Solution n°11

On commence par créer (en `root`) les dossiers et fichiers.

```
1 mkdir /home/joe/admin
2 touch /home/joe/README /home/joe/admin/secret.txt
```

Puis on attribue les bons propriétaires et groupes.

```
1 chown joe:joe /home/joe/README
2 chown joe:administrateurs /home/joe/admin
3 chown joe:administrateurs /home/joe/admin/secret.txt
```

Enfin on configuré correctement les droits d'accès.

```
1 chmod 644 /home/joe/README
2 chmod 770 /home/joe/admin
3 chmod 660 /home/joe/admin/secret.txt
```

On peut ensuite observer la bonne configuration des droits.

```
1 root@europa:~# ls -l /home/joe/README
2 -rw-r--r-- 1 joe joe 0 4 déc. 16:10 /home/joe/README
3 root@europa:~# ls -la /home/joe/admin
4 total 8
5 drwxrwx--- 2 joe administrateurs 4096 4 déc. 16:10 .
6 drwxr-xr-x 3 joe joe 4096 4 déc. 16:10 ..
7 -rw-rw---- 1 joe administrateurs 0 4 déc. 16:10 secret.txt
```

## Solution n°12

[exercice p. 24]

### Exercice

Qu'est-ce que le groupe primaire d'un utilisateur Linux ?

**A** Le dernier groupe dans lequel il vient d'être ajouté

**B** Le groupe principal dans lequel il se trouve, généralement attribué à la création de l'utilisateur

**C** Le groupe qu'il partage avec l'utilisateur root

### Exercice

Qu'est-ce qu'un UID ?

**A** Un identifiant unique associé à la machine

**B** Un identifiant associé à un groupe d'utilisateur

**C** Un identifiant associé à un utilisateur

### Exercice

Par défaut il est conseillé d'utiliser tout le temps l'utilisateur root pour administrer la machine.

**A** Vrai

**B** Faux



Au contraire, il vaut mieux n'utiliser les privilèges de super-utilisateur que lorsque c'est nécessaire.

**Exercice**

Dans quel fichier est-il possible de trouver la configuration des utilisateurs de la machine ?

**A** /etc/passwd

**B** /root/passwd

**C** /root/users

**D** /etc/users

**Solution n°13**

[exercice p. 25]

**Exercice**

Pour créer un groupe sur Linux, on préfère :

**A** utiliser la commande `addgroup`

**B** utiliser la commande `groupadd`

**C** modifier le fichier `/etc/group`



- On évite généralement de modifier directement le fichier `/etc/group`.
- La commande `addgroup` est plus facile à utiliser, mais elle n'est pas disponible sur tous les Linux.
- La commande `groupadd` est moins ergonomique mais elle est toujours disponible.

**Exercice**

Pour lancer une commande avec les privilèges super-utilisateur, quelles sont les différentes possibilités ?

**A**

Me connecter avec `su` n'est pas une pratique conseillée, mais certaines installations permettent au compte `root` de se connecter au démarrage.

**B**

Utiliser la commande `sudo -i` pour ouvrir un terminal `root` puis lancer ma commande

**C** Lancer la commande `sudo MACOMMANDE`

**D** Utiliser la commande `root MACOMMANDE`

### Exercice

Si je souhaite donner l'accès à un fichier à plusieurs utilisateurs précis.

**A** Il faut donner des droits en lecture à tout les utilisateurs du système

**B** Ce n'est pas possible

**C** Il faut ajouter les utilisateurs en question dans un groupe et donner les droits de lecture de notre fichier à ce groupe

## Solution n°14

[exercice p. 25]

### Exercice

Quelles sont les affirmations vraies pour un fichier dont `ls -l` affiche les informations suivantes ? `drwxrwxr-x`

**A** Il s'agit d'un dossier

**B** Seul le propriétaire a des droits d'écriture

**C** Tout les utilisateurs peuvent le lire

**D** Il n'est associé à aucun groupe

 Le caractère "d" au début indique que c'est un dossier, et on constate que le groupe a les droits `rwx`, donc qu'il n'y a pas que le propriétaire qui a des accès en écriture. Les droits pour tout les utilisateurs `r-x` indique que le dossier est accessible à tout le monde.

### Exercice

Quels sont les droits que l'on s'attend à avoir si l'on fait la commande `chmod 640 ?`

**A** `rw-rw----`

**B** `rw-rw----`

**C** `r-xr-----`

**D** rw-r-----

🔍 Si l'on décompose les droits appliqués

- 6 signifie lecture et écriture (4+2) donc rw- pour le propriétaire
- 4 signifie lecture donc r - - pour le groupe
- 0 signifie aucun droits donc - - - pour les autres utilisateurs

# Contenus annexes

## 1. Commandes de bases sous Linux : cd, ls, mkdir, rm, find, cat, nano...

### Le manuel !

La commande `man` permet d'afficher le manuel d'une autre commande.

Par exemple `man ls` permet d'afficher les options de la commande `ls`, et `man man` est l'affichage du manuel du manuel.

### Organisation des fichiers

Les fichiers sont organisés sous Linux dans des dossiers (ou répertoires) arborescents (il n'y a pas de notion de disques).

Le premier de ces dossiers, appelé racine, est `/`.

Des fils courants de racines sont :

- `/bin` qui contient des programmes du système ;
- `/home` qui contient les données des utilisateurs ;
- `/tmp` qui contient des données volatiles accessibles à tous.

### Gestion de fichiers

- `pwd` savoir où je me situe dans l'arborescence
- `cd` aller quelque part dans l'arborescence
  - `cd /home` aller dans `/home` (déplacement absolu)
  - `cd ..` aller dans le répertoire parent de mon répertoire courant (déplacement relatif)
  - `cd ..` remonter dans l'arborescence (déplacement vers son père)
  - `cd ~` permet de retourner dans son dossier initial (retour au domicile)
- `ls` voir les fichiers et dossiers dans mon dossier actuel (sauf les fichiers cachés commençant par un `.`)
- `ls -al` voir tous les fichiers et dossiers dans mon dossier actuel avec leurs informations associées (vue détaillée)
- `touch file` créer un fichier `file`
- `rm file` supprimer le fichier `file` dans mon dossier actuel
- `mkdir dir` créer un nouveau dossier `dir` dans mon dossier actuel
- `rm *` supprimer tous les fichiers de mon dossier actuel
- `rm -R dir` supprimer le dossier `dir` dans mon dossier actuel
- `cat file` afficher le contenu du fichier `file`
- `more` afficher le contenu du fichier `file` en mode paginé
- `less` afficher le contenu du fichier `file` en mode défilement

## Édition de fichiers

- nano file
  - Éditeur dans le terminal (simple d'utilisation)
  - Les commandes sont indiqués en bas de l'éditeur
- gedit file &
  - Éditeur graphique
  - Utiliser une extension de fichier standard ou le menu Affichage > Mode de coloration pour obtenir une visualisation adapté au type de fichier édité (par exemple .sql pour un fichier SQL)

## Rechercher un fichier

- `find / -name '*test*'`  permet de rechercher un fichier contenant la chaîne test sur tout le disque
- `find ~ -name '*test*'`  permet de rechercher un fichier contenant la chaîne test dans son espace personnel

 Complément

*Agir en tant que root (su / sudo) (cf. p.34)*

*Installer des applications sous Linux (cf. p.35)*

## 2. Agir en tant que root (su / sudo)

Un système Linux est généralement utilisé par plusieurs utilisateurs (ou *user*) identifiés par un compte.

Tous les système ont en commun le premier de ces utilisateurs : *root*, l'administrateur du système qui possède tous les droits.

### su

 Syntaxe

Pour se connecter en tant qu'utilisateur *root* (et pouvoir faire les opérations qui lui sont réservées) : `su root`.

- Le mot de passe de utilisateur *root* est demandé par le système.

### sudo

 Syntaxe

Pour exécuter une commande en tant que *root* sans changer d'*user* : `sudo commande`.

- Le mot de passe de utilisateur lançant `sudo` est demandé par le système.
- Seuls certains utilisateurs appelés *sudoers* ont le droit d'exécuter la commande `sudo`.

- Pour ajouter un utilisateur dans la liste des *sudoers*, exécuté en tant que *root* (ou via un autre utilisateur *sudoers*) :
  - `adduser user sudo` (en tant que *root*)
  - `sudo adduser user sudo` (avec un utilisateur déjà membre des *sudoers*)

### Debian (sudo)

⚠ Attention

Sous Debian la commande `sudo` n'est pas installée par défaut.

### Ubuntu (su)

⚠ Attention

Sous Ubuntu l'utilisateur *root* n'est pas actif et il n'est donc pas possible d'exécuter la commande `su root`. On peut en revanche utiliser `sudo`, l'utilisateur créé à l'installation est membre des *sudoers*.

Pour activer le compte *root*, il faut lui attribuer un mot de passe avec la commande : `sudo passwd root`.

### Agir en tant qu'un autre utilisateur

⊕ Complément

Pour agir en tant qu'utilisateur *user*, exécuter :

`su user`, ou `su - user` pour adopter l'environnement de *user*.

## 3. Installation d'applications sous Debian et Ubuntu (apt-get)

Pour installer une application exécuter la commande suivante en tant que *root* ou via *sudo* :

```
sudo apt-get install applications
```

### Installer virtualbox, lftp, cups-pdf, pdftk, dia, gimp et git

👁 Exemple

```
1 sudo apt-get install virtualbox lftp cups-pdf pdftk dia gimp git
```

### Maintenir le système à jour

⊕ Complément

Pour mettre le système à jour, exécuter les deux commandes suivantes en tant que *root* ou via *sudo* :

1. `sudo apt-get update`
2. `sudo apt-get upgrade`

