Introduction aux systèmes d'exploitation - M2 CCI

Utilisateurs et droits sous Unix

Nicolas Louvet (Univ. Lyon 1)

d'après un cours pompé sur Laure Gonnord
qui avait pompé elle-même sur Julien Forget

1 Utilisateur, groupe
- Commandes utilisateur utiles
- Commandes administrateur utiles

2 Détour sur les fichiers et les répertoires

3 Droits d'accès

Généralités

Sous Unix, un utilisateur possède :

- Un nom `login` qui est associé à un numéro `id`
- Un mot de passe (password)
- Un espace de stockage dédié `/home/login/`

Il fait aussi partie d'un groupe d'utiliseurs (qui a un numéro : `gid`.)

1. Utilisateur, groupe
   - Commandes utilisateur utiles
   - Commandes administrateur utiles
2. Détour sur les fichiers et les répertoires
   - Fichiers
   - Répertoires
3. Droits d'accès
   - Notion de droit
   - Donner des droits

Password

Le mot de passe :

- peut être changé par la commande `passwd`
- **NE PAS UTILISER** un mot du dictionnaire (pas seulement français !)
- **NE PAS UTILISER** votre nom ou prénom, ni d'un parent, ...
- **NE PAS UTILISER** une information vous concernant (numéro de téléphone, date d'anniversaire, ...)
- **UTILISER** une combinaison de majuscules, minuscules, chiffres, caractères spéciaux et au moins 12 caractères.

Pour changer d'utilisateur

`su` (substitute user)

`su [-] [nom [argument]]`

- permet de changer d'utilisateur sans se "déloguer" puis se "reloguer"
- la commande `su` sans paramètre permet de passer en mode super-utilisateur
- `exit` permet de revenir à l'utilisateur d'origine

```
jm@p166bis jm]# pwd
/home/jm
[jm@p166bis jm]$ su
Password:
[root@p166bis jm]# pwd
/home/jm
[root@p166bis jm]# exit
exit
[jm@p166bis jm]$
```

```
[jm@p166bis jm]# pwd
/home/jm
[jm@p166bis jm]$ su -
Password:
[root@p166bis /root]# pwd
/root
[root@p166bis /root]# exit
logout
[jm@p166bis jm]$
```

Avoir des infos sur les autres

Pour savoir qui est connecté sur la machine :

- `who [-options]`
- `users`

```
jduthill@weppes:~ >who
jduthill      pts/0          Mar  1   08:39 (douaisis.escaut.net)
mcastele      pts/1          Feb 24   14:31 (172.26.70.218)
dvivier       :0             Feb 28   14:01
ocaron        pts/2          Feb 27   11:00 (baleares.staff.yser.net)
jduthill@weppes:~ >users
jduthill mcastele dvivier ocaron
jduthill@weppes:~ >
```

Autres infos

Connaître l'`uid` et le `gid` d'un usager :

`id [login]`

```
jduthill@douaisis:~$ cat /etc/passwd | grep duthill
jduthill:x:1127:1011:Jean-Michel Duthilleul:/home/imaEns/jduthill:/bin/bash
rduthill:x:1475:1136:R
```

```
jduthill@douaisis:~$ cat /etc/group | egrep "duthill|1011"
sinfo::1059:nom1,nom2,jduthill,nom3,nom4,
gisEns::1046:user1, user2,user3,jduthill,user4,user5,user6
imaEns::1011:User1, User2,User3,
```

```
jduthill@douaisis:~$ su
Password:
douaisis:/home/imaEns/jduthill# id
uid=0(root) gid=0(root) groups=0(root)
douaisis:/home/imaEns/jduthill# id jduthill
uid=1127(jduthill) gid=1011(imaEns) groups=1011(imaEns),1059(sinfo),1046(gisEns)
douaisis:/home/imaEns/jduthill#
```

1. Utilisateur, groupe
   - Commandes utilisateur utiles
   - Commandes administrateur utiles
2. Détour sur les fichiers et les répertoires
   - Fichiers
   - Répertoires
3. Droits d'accès
   - Notion de droit
   - Donner des droits

Ajout d'un utilisateur

La commande bas-niveau :

```
useradd [-d home_dir] [-g group] [-p passwd] [-s shell] [-u
                        uid] login
```

- `-d home_dir` : répertoire d'accueil au login de l'utilisateur (`$HOME`)
- `-g group` : nom ou numéro du groupe auquel appartient l'usager. Ce groupe doit exister.
- `-p passwd` : le mot de passe de l'utilisateur
- `-s shell` : le nom du shell qui sera utilisé au login de l'usager.
- les informations sont rangées dans le fichier `/etc/passwd` (fichier en lecture pour tous) sauf le mot de passe qui est stocké dans le fichier `/etc/shadow` (en lecture pour root)

Heureusement, il y a plus simple :

```
adduser login
```

Cette commande fonctionne en mode intéractif, et est plus facile à utiliser !

Ajout d'un groupe

`addgroup | groupadd [-g gid] group`

- `-g gid` : c'est la valeur numérique de l'identifiant de groupe (GID).
  Il doit être UNIQUE.
  En général, les valeurs comprises entre 0 et 499 sont réservées au système.
  Par défaut ce sera la première valeur supérieure à tous les autres groupes.
- les informations sont rangées dans le fichier `/etc/group`

Gestion des utilisateurs

fichiers `/etc/passwd`, `/etc/shadow` et `/etc/group`

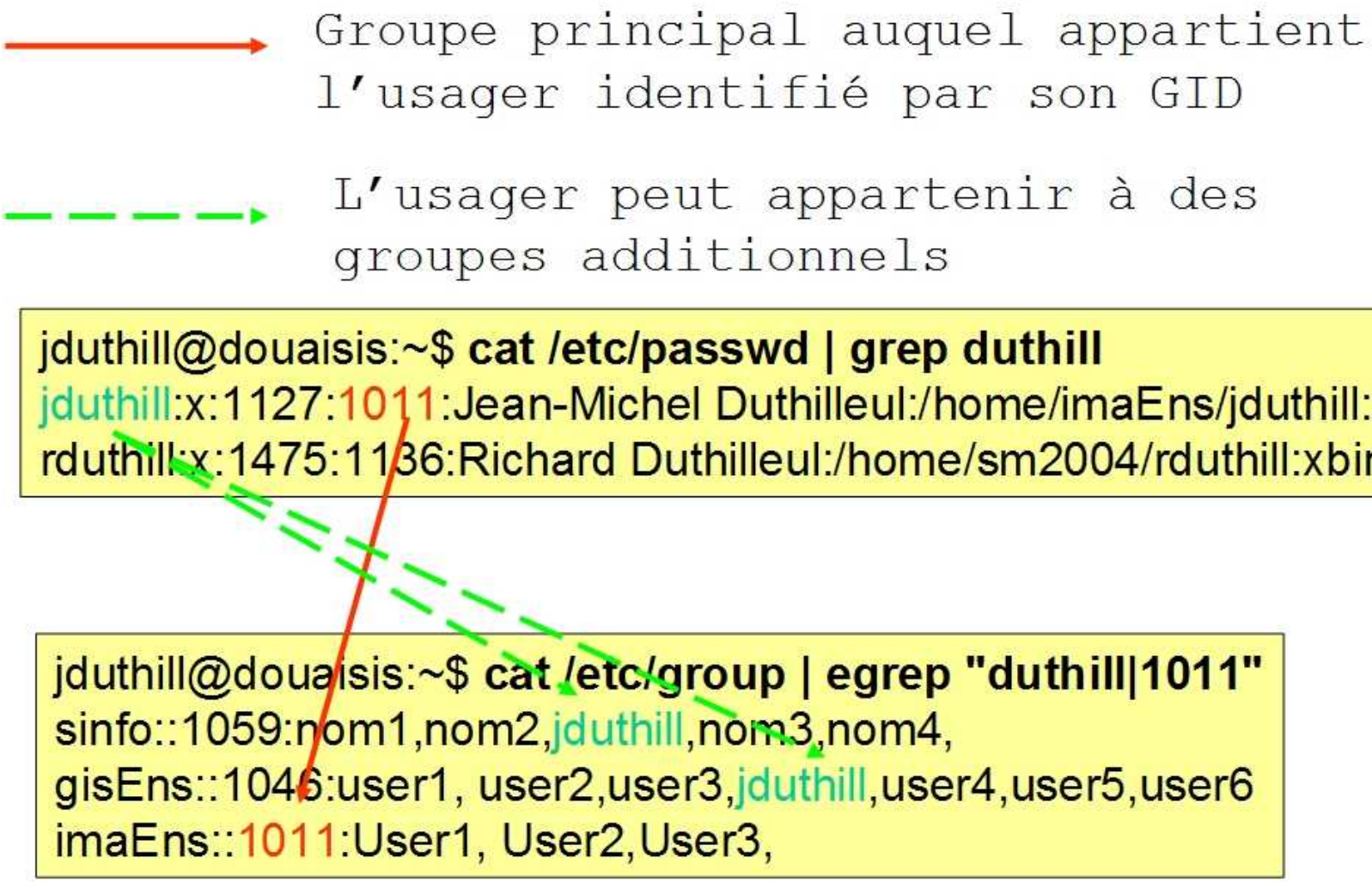
```
[root@p166bis /etc]# cat group
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon
tty:x:5:
disk:x:6:root
lp:x:7:daemon,lp
jm:x:503:
```

```
[root@p166bis jm]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
jm:x:500:503::/home/jm:/bin/bash
```

```
[root@p166bis jm]# cat /etc/shadow
root:$1$AaYrU0.H$v84TWFOOdJGo3TM.RVL1z.:11006:0:99999:7:::
bin:*:11006:0:99999:7:::
daemon:*:11006:0:99999:7:::
adm:*:11006:0:99999:7:::
lp:*:11006:0:99999:7:::
jm:$1$XQ6CQ60R$5hKGfOQqptQzM27/OZT1p0:11011:0:99999:7:::
```

Gestion des utilisateurs

fichiers `/etc/passwd` et `/etc/group`



Gestion des utilisateurs

fichiers `/etc/passwd` et `/etc/group`

```
jduthill@douaisis:~$ cat /etc/passwd | grep duthill
jduthill:x:1127:1011:Jean-Michel Duthilleul:/home/imaEns/jduthill:/bin/bash
rduthill:x:1475:1136:Richard Duthilleul:/home/sm2004/rduthill:xbinxbash
```

```
login:x:uid:gid:commentaire:répertoire_accueil:shell
```

```
jduthill@douaisis:~$ cat /etc/group | egrep "duthill|1011"
sinfo::1059:nom1,nom2,jduthill,nom3,nom4,
gisEns::1046:user1, user2,user3,jduthill,user4,user5,user6
imaEns::1011:User1, User2,User3,
```

```
groupe::gid:[user,user,….,]
```

Suppression

- Suppression d'un utilisateur : `deluser login`
- Suppression d'un groupe : `delgroup group`
- Retirer un utilisateur d'un group : `deluser login group`

Bref, ne vous souviendrez jamais de tout ça, faites `man deluser` !

Petit résumé

- Ajout d'un nouvel utilisateur : `adduser cyan`
  Par défaut son groupe principal est `cyan`
  Cette commande crée aussi son répertoire d'accueil `/home/cyan`.

- Ajout d'un nouveau groupe : `addgroup bleu`

- Ajout d'un groupe existant à un utilisateur : `adduser cyan bleu`

- Ajout d'un groupe comme *groupe principal* :
  `usermod -g bleu cyan`

- Ajout d'un nouvel utilisateur avec son groupe principal :
  `adduser cyan --ingroup bleu`

- Attention, lors de la suppression d'un utilisateur, son répertoire d'accueil n'est pas automatiquement supprimé.

1. Utilisateur, groupe

2. Détour sur les fichiers et les répertoires
   - Fichiers
   - Répertoires

3. Droits d'accès

Fichiers

Definition

Un fichier est un ensemble d'informations réunies sous un même nom et enregistrées sur un support de stockage à long terme.

Ils permettent :

- Le stockage des informations calculées par un processus lorsqu'il se termine ;
- Le stockage des informations de grande taille ;
- L'accès de plusieurs processus à ces informations.

Exemple d'informations à stocker :

- Fichier de *données* (vidéo, image, base de données, etc).
- Fichier *texte* : caractères organisés en lignes/pages.
- Fichier *source* : fichier texte constitué de fonctions compilables.
- Fichier *exécutable* : code que le système peut exécuter.

Fichier = une séquence de bits, octets, dont le sens est défini par une convention entre le créateur et l'utilisateur.

Nom et attributs

- Un fichier est nommé pour notre commodité.
  - Le nom est généralement une chaîne de caractères ;
  - Le nom peut contenir une extension indiquant son type ;
  - Certains OS font une différence entre majuscules et minuscules.

- Les attributs varient selon les systèmes de gestion de fichiers.
  - Type (si le système supporte différents types de fichiers) ;
  - Emplacement : pointe sur le fichier ;
  - Taille : taille courante du fichier (octets, mots ou blocs) ;
  - Droits d'accès : qui peut lire, écrire, exécuter, etc ;
  - Date et heure de création, dernière modification, dernière utilisation...

- Ces informations sont généralement rangées dans la structure de répertoire qui réside également sur le support.

Types de fichiers

UNIX

- - : ordinaire (texte, binaire, exécutable, ...)
- d : répertoire (*directory*)
- c : *fichier périphérique* (*device*) de type caractère
- b : *fichier périphérique* (*device*) de type bloc
- s : socket
- l : lien (*link*)
- p : tube (*pipe*) nommé

▶ Ne retenir que les deux premiers.

Liens durs

Un fichier est en fait une place sur le disque dur, référencée par un numéro...

- lien dur (*hard link*)
  - autre nom pour un fichier
  - un lien de ce type peut être détruit sans affecter les autres
  - dès qu'un fichier n'a plus de liens, il est détruit

```
jduthill@weppes:~/tp >ls -li essailien
556622253 -rw-r--r--    1 jduthill imaEns        547 Apr  3 15:31 essailien
jduthill@weppes:~/tp >cd
jduthill@weppes:~ >ln tp/essailien ressailien
jduthill@weppes:~ >ln tp/essailien ressailien2
jduthill@weppes:~ >ls -li ressailien*
556622253 -rw-r--r--    3 jduthill imaEns        547 Apr  3 15:31 ressailien
556622253 -rw-r--r--    3 jduthill imaEns        547 Apr  3 15:31 ressailien2
jduthill@weppes:~ >rm tp/essailien
jduthill@weppes:~ >ls -li ressailien*
556622253 -rw-r--r--    2 jduthill imaEns        547 Apr  3 15:31 ressailien
556622253 -rw-r--r--    2 jduthill imaEns        547 Apr  3 15:31 ressailien2
```

Liens symboliques

- lien symbolique (*soft link*)
  - c'est un pointeur sur un fichier
  - si l'original est détruit, le lien persiste mais ne fonctionne plus
  - un lien sur un répertoire doit être un lien symbolique

```
jduthill@weppes:~/tp >ls -li essailien
556621998 -rw-r--r--   1 jduthill imaEns   547 Apr  3 15:37 essai
jduthill@weppes:~/tp >cd
jduthill@weppes:~ >ln -s tp/essailien ressailien
jduthill@weppes:~ >ls -li ressailien
555788376 lrwxrwxrwx   1 jduthill imaEns    12 Apr  3 15:38 ressai -> tp/essai
jduthill@weppes:~ >rm tp/essailien
jduthill@weppes:~ >cat ressailien
cat: ressailien: No such file or directory
jduthill@weppes:~ >ls -li ressailien
555788376 lrwxrwxrwx   1 jduthill imaEns    12 Apr  3 15:38 ressai -> tp/essai
```

Opérations sur les fichiers

- Créer un fichier :
  - Trouver de la place dans le système de fichiers
  - Créer une entrée dans le répertoire le contenant
  - L'entrée du répertoire mémorise le nom du fichier et son emplacement
- Écrire dans un fichier :
  - Rechercher le fichier (retrouver le répertoire puis l'emplacement disque)
  - Maintenir un pointeur d'écriture vers l'emplacement où l'écriture suivante aura lieu
- Lire dans un fichier :
  - Rechercher le fichier (retrouver le répertoire puis l'emplacement disque)
  - Maintenir un pointeur de lecture vers l'emplacement de la lecture suivante
- Détruire un fichier :
  - Rechercher le fichier
  - Libérer tout l'espace disque occupé par ce fichier
  - Supprimer l'entrée dans le répertoire

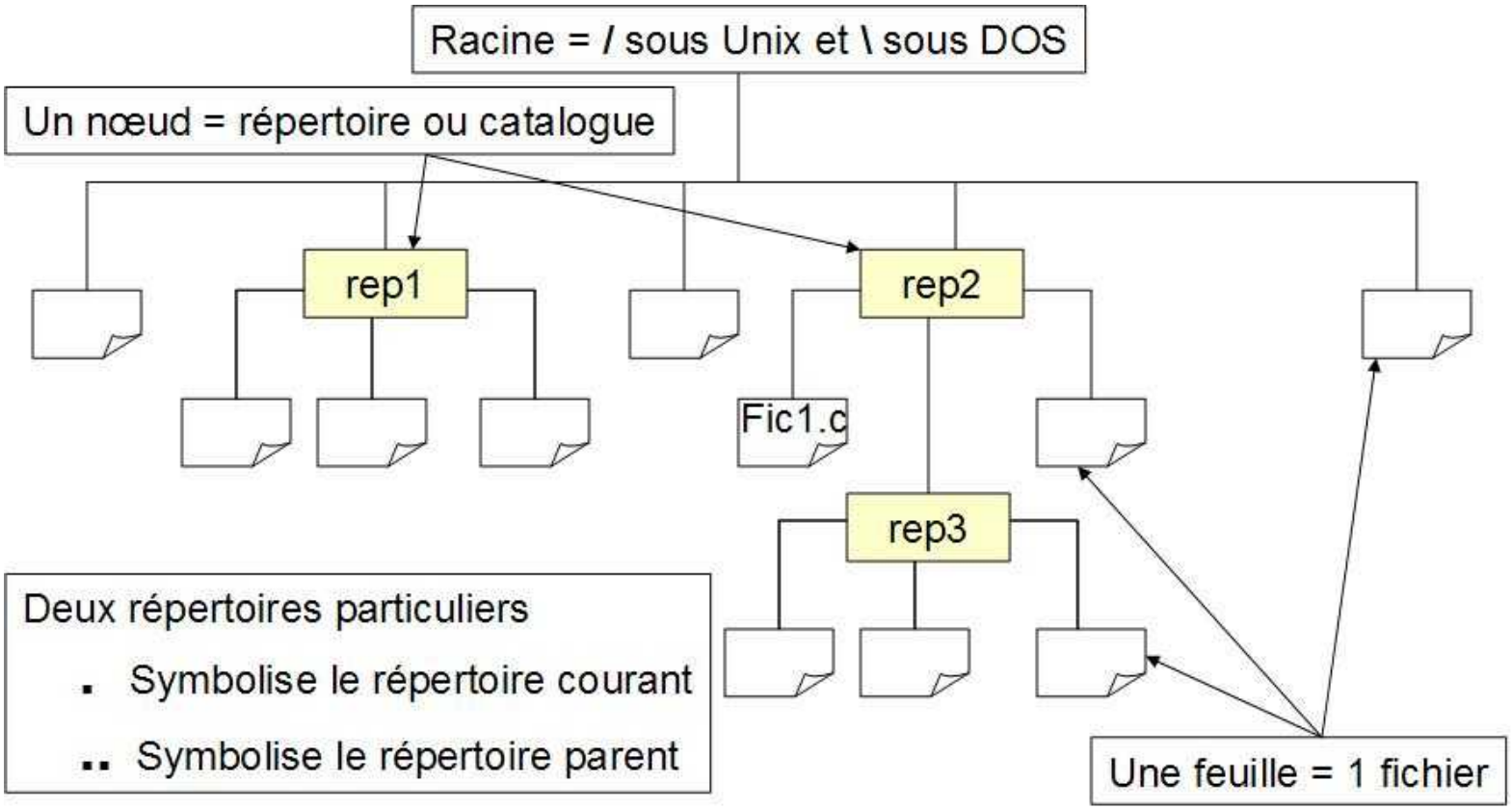
Définition

Definition

Un répertoire est une liste de descriptions de fichiers.

- Objectif :
  - Stockage de milliers de fichiers ;
  - Organisation des données.
- Mise en œuvre :
  - Un répertoire mémorise les noms, attributs et emplacements sur le disque des fichiers ;
  - Un répertoire est le plus souvent lui-même un fichier ;
  - Les répertoires peuvent en général contenir d'autres répertoires, formant ainsi une structure arborescente, qui permet aux utilisateurs de créer leurs propres sous-répertoires et d'organiser leurs données.

Exemple d'arborescence - Rappel



UNIX : principaux répertoires

- / : la racine
- `bin` : commandes binaires utilisateurs essentielles
- `boot` : fichiers statiques du chargeur de lancement
- `dev` : fichiers de périphériques
- `etc` : configuration système de la machine
- `home` : répertoires personnels des usagers
- `lib` : bibliothèques partagées essentielles
- `mnt` : point de montage temporaire
- `proc` : fichiers virtuels d'information du noyau et des processus
- `root` : répertoire personnel de `root`
- `sbin` : binaires systèmes
- `tmp` : fichiers temporaires
- `usr` : deuxième section majeure du système
- `var` : contient des fichiers de données variables

Disques dans l'arborescence

Sous Unix

- Pas de disque dans les chemins ;
- Un arbre unique de répertoires, de racine "/" ;
- Disques vus comme des *fichiers spéciaux* ;
- Contenu d'un fichier disque présenté sous forme d'arborescence dans un répertoire donné par la commande `mount`.

```
laure@sorlin:~$ df -k
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda3            226322844  20330896  194488692  10% /
none                         4         0          4   0% /sys/fs/cgroup
udev                   4020328         4    4020324   1% /dev
tmpfs                   805976       916     805060   1% /run
none                      5120         0       5120   0% /run/lock
none                   4029864       484    4029380   1% /run/shm
none                    102400        36     102364   1% /run/user
```

1. Utilisateur, groupe
2. Détour sur les fichiers et les répertoires
3. Droits d'accès
   - Notion de droit
   - Donner des droits

Notion de droit

Le système de fichiers fournit un mécanisme de gestion des droits d'accès, pour :

- connaître le propriétaire d'un fichier / répertoire
- sécuriser l'accès aux fichiers (vie privée).

Definition

Les droits sont un ensemble de règles qui régissent les *permissions* d'accès à certains fichiers/répertoires.

NB : Le mécanisme interne de gestion des droits des fichiers (où est stockée l'info) ne sera pas expliqué ici.

ugo

Pour un fichier/un répertoire, on fait la distinction entre :

- l'utilisateur propriétaire (u)
- le groupe propriétaire (g)
- les autres (o)

En général l'utilisateur propriétaire est celui qui a créé le fichier, et le groupe propriétaire est son groupe.

rwx

Lorsqu'on fait

```
laure@meert:~$ ls -l notes_soutenances.ods
-rw-rw-r-- 1 laure laure 22701 aout  29 14:40 notes_soutenances.ods
laure@meert:~$ ls -l documents
total 156560
drwxr-xr-x  9 laure laure      4096 aout   3  2009 0809work
...
```

Les droits apparaissent comme première info à gauche

- 1 info fichier ou répertoire : - ou d.
- 3 triplets : `r` pour read, `w` pour write, `x` pour execute, dans l'ordre UGO.
- "tiret" signifie que le droit n'est pas accordé.

rwx : des détails

Les droits s'attribuent sur trois critères :

- **la lecture (r) :**
  - on peut par exemple lire le fichier avec un logiciel
  - pour un répertoire, il autorise l'affichage de son contenu (`ls`)
- **l'écriture (w) :**
  - on peut modifier le fichier et le vider de son contenu
  - pour un répertoire, il autorise création/suppression/changement de nom des fichiers qu'il contient, (quels que soient les droits d'accès des fichiers de ce répertoire, même s'ils ne possèdent pas eux-mêmes le droit en écriture)
- **l'exécution (x) :**
  - on peut exécuter le fichier s'il est exécutable
  - pour un répertoire, il autorise l'accès au répertoire, on peut s'y rendre (`cd`)

Attention, pour `ls` il faut `rx` pour le répertoire pour pouvoir accéder au détails sur les fichiers !

En fait, il existe plein d'autres détails sordides...

Donner des droits

- `chmod` pour modifier `rwx` pour `ugo`. Exemples :
  `chmod u+rw fichier.txt` donne à l'utilisateur (soi-même) les droits lecture+écriture sur le fichier texte.

  `chmod g+x script.sh` donne à son groupe les droits en exécution sur le fichier (ici un script shell).

- `chown` pour modifier le propriétaire d'un fichier. Exemple :
  `chown toto SysCours` change le propriétaire du répertoire `SysCours` vers l'utilisateur `toto`.
  - Commande possible par le propriétaire (ou root).

Notion de masque

Le paramètre système définissant les droits accordés par défaut s'appelle **le masque utilisateur** :

- Dans Ubuntu, le masque utilisateur par défaut accorde les permissions `rw-r-r-` pour les fichiers, et `rwxr-xr-x` pour les répertoires.
- Peut être changé pour une session, avec `umask`, ou de manière permanente, ou dans des fichiers de config (`/etc/profile`).

Limite des droits.

Soit un script shell situé dans `/home/laure/bin/`. Comment donner les droits d'exécution au script shell nommé `toto.sh` à l'utilisateur `stephane` ?

- solution simple si `stephane` est dans le même groupe
- solution plus complexe pour autoriser un utilisateur unique
  ↪ avec "les droits fins" : ACL