Leçon 3 : Administration du disque dur

3.1 Les entrées dans le système de fichiers

Sous Unix, il existe plusieurs types d'entrée différente. La commande ls permet de montrer les détails concernant ces « fichiers » particuliers. Le 1ier caractères désigne le type d'entrée. Ainsi, on trouve :

- Des *fichiers* (`-`) ou des *dossiers* (`d`) : `ls -l /etc`, montre notamment :
  ```
  -rw-r--r--.   1 root root      970 Mar  9  2015 yum.conf
  drwxr-xr-x.   2 root root     4096 Jul 28 11:58 yum.repos.d
  ```
- Des *liens symboliques* (`l`) : `ls -l /bin`
  ```
  lrwxrwxrwx. 1 root root 7 Jul 27 00:55 /bin -> usr/bin
  ```
- Des *périphériques blocs* (`b`) ou *caractères* (`c`): `ls -l /dev/sda /dev/tty`
  ```
  brw-rw---- 1 root disk 8, 0 Sep 15 22:16 /dev/sda
  crw-rw-rw- 1 root tty  5, 0 Sep 15 22:05 /dev/tty
  ```
- Des *sockets locaux* (`s`) : `ls -l /dev/log`
  ```
  srw-rw-rw- 1 root root 0 Sep 15 22:05 /dev/log
  ```

3.2 Les liens

Les liens sont des entrées particulières qui permettent de *pointer* vers un autre endroit du disque dur. Ainsi, un *lien symbolique* permet de créer une entrée pointant vers un dossier ou un fichier ailleurs. L'intérêt principal étant de faire apparaître, dans des dossiers différents, les mêmes entrées.

Attention ! Il ne s'agit pas d'une copie mais bien d'un *pointeur*. Quand on ouvre un lien, c'est l'élément pointé qui est ouvert.

3.3 Les droits

Les droits dans les systèmes UNIX sont de différents types : il y a les *droits classiques* et les *ACLs*.

3.3.1 Les droits classiques

Les droits classiques sont ceux qui sont affichés lorsqu'on exécute la commande ls -l. Nous pourrions obtenir le résultat suivant sur le dossier /home :

```
$ ls -l /home
total 28
drwx------.  2 root  root  16384 Jul 27 00:54 lost+found
drwx------. 15 lsw   users  4096 Aug 13 22:12 lsw
drwxrwxr-x+  2 root  root   4096 Aug 11 23:27 public
drwx------  18 swila users  4096 Aug 14 00:01 swila
```

Nous y voyons des combinaisons étranges des lettres `r`, `w`, `x` et du symbole `-`. Les permissions représentées sont : *l'autorisation en lecture (`r`), l'autorisation en écriture (`w`) et l'autorisation en exécution / accès (`x`).*

Ces permissions sont présentes à la fois pour les dossiers ou les fichiers. Elles sont présentes sous la forme d'un trio, répété trois fois (ex : `rwxrwxr-x`).

Chaque trio doit être interprété comme suit :

Permissions	Explication
rwx	Autorisation en lecture, en écriture, en accès / exécution

r-x	Autorisation en lecture, en exécution / accès
r--	Autorisation en lecture
--x	Autorisation en exécution / accès
rw-	Autorisation en lecture, en écriture

Ainsi, la présence de la lettre dans le trio indique que l'autorisation correspondante est présente alors qu'un tiret (`-`) indique que l'autorisation correspondante est absente.

Comme le trio est répété trois fois, cela indique que les permissions s'adressent à des groupes d'utilisateurs différents. Ainsi dans l'exemple du dossier `/home` présenté ci-avant, nous avons :

```
drwx------. 15 lsw   users  4096 Aug 13 22:12 lsw
```

Le premier trio (`rwx`) indique que les autorisations mentionnées sont accordées **à l'utilisateur propriétaire** qui dans notre cas, est `lsw`. Le second trio (`---`) indique que les autorisations mentionnées (aucun droit ici) sont accordées **au groupe propriétaire** qui dans cet exemple est `users`. Enfin, le dernier trio (`---`) indique que les autorisations mentionnées (aucun droit ici) sont accordées **aux autres utilisateurs** *(i.e. qui ne sont ni l'utilisateur propriétaire, ni membres du groupe propriétaire)*.

```
drwxrwxr-x+  2 root  root   4096 Aug 11 23:27 public
```

Le premier trio (`rwx`) indique que les autorisations mentionnées sont accordées **à l'utilisateur propriétaire** qui dans notre cas, est `root`. Le second trio (`rwx`) indique que les autorisations mentionnées sont accordées **au groupe propriétaire** qui dans cet exemple est `root`. Enfin, le dernier trio (`r-x`) indique que les autorisations mentionnées sont accordées **aux autres utilisateurs** *(i.e. qui ne sont ni l'utilisateur propriétaire, ni membres du groupe propriétaire)*.

Le symbole `+`, présent à la fin des permissions, informe que des ACLs sont définies également (voir plus loin).

Les permissions sur les fichiers

L'autorisation en lecture (`r`) sur un fichier indique *qu'il est permis de lire le contenu du fichier*. L'autorisation en écriture (`w`) sur un fichier indique *qu'il est permis de modifier le contenu du fichier*. L'autorisation en exécution (`x`) sur un fichier indique *que ce fichier peut être exécuté*.

Les permissions sur les dossiers

L'autorisation en lecture (`r`) sur un dossier indique qu'il est permis de lire le contenu du répertoire (les fichiers qui s'y trouvent). L'autorisation en écriture (`w`) sur un dossier indique qu'il est permis de modifier le contenu du répertoire (ajouter / supprimer des fichiers ou dossiers à l'intérieur). Enfin, l'autorisation en exécution (`x`) sur un dossier indique qu'il est permis de traverser le répertoire.

Un exemple particulier :

```
drwx--x--x. 15 lsw   users  4096 Aug 13 22:12 lsw
```

Dans cet exemple, le dossier `lsw` peut être lu, modifié et traversé par l'utilisateur `lsw`. Par contre, les membres du groupe `users` et les autres peuvent seulement traverser (i.e. entrer dans) le répertoire sans pouvoir lire son contenu.

Cette particularité est parfois intéressante pour *donner une autorisation plus grande sur un dossier à l'intérieur*.

Modification des droits

La modification des *droits classiques* est assez simple. Cette modification peut prendre à la fois la forme d'un changement de permission ou d'un changement de propriétaire (utilisateur ou groupe propriétaire). Les commandes suivantes sont utilisées :

chmod

La commande `chmod` permet de modifier les permissions sur un dossier ou un fichier. Pour exprimer la permission souhaitée, il est possible d'utiliser une forme symbolique ou numérique.

La méthode numérique consiste à interpréter le trio `rwx` sous une forme binaire. Ainsi le droit `r-x` se traduit, en binaire par `101`, c'est-à-dire `5`. Ecrire `755` représente la permission `rwxr-xr-x`.

La méthode textuelle consiste à décrire les doits souhaités en utilisant les raccourcis suivants : `u` pour l'utilisateur propriétaire, `g` pour le groupe propriétaire et `o` pour les autres. Ainsi, écrire `u=rwx,g=rx,o=-` représente la permission `rwxr-x---`.

```
$ chmod 755 mondossier
$ chmod u=rwx,go=rx mondossier
```

chown

La commande `chown` permet de changer l'utilisateur et le groupe propriétaire d'un dossier ou fichier. Par exemple :

```
$ chown lsw:users mondossier
```

chgrp

La commande `chgrp` permet de modifier le groupe propriétaire d'un dossier ou d'un fichier.

3.4 Les ACLs

La limitation des droits classiques a conduit à l'introduction des *ACLs*. Une ACL est un droit spécifique, mentionnant des permissions pour un utilisateur ou un groupe donné. L'ACL est attachée à un dossier ou un fichier. Dans certains cas, l'ACL est transmise aux dossiers ou fichiers contenus.

Ainsi, grâce à une ACL il est possible de préciser que l'utilisateur `lsw` peut lire ou modifier un fichier donné.

Bien sûr, il est possible d'attacher *plusieurs ACLs* à un fichier ou dossier pour permettre ainsi à l'utilisateur `lsw` mais aussi à l'utilisateur `swila` d'accéder au dossier, par exemple. Les ACLs doivent être combinées avec les droits classiques.

Certains systèmes de fichier nécessite d'activer le support des ACLs. Par défaut sur les distributions CentOS 7, les ACLs sont actives sur les systèmes de fichier *ext4* et *xfs*. Dans notre cas, cela signifie que les ACLs peuvent être utilisées n'importe où.

3.4.1 Les ACLs classiques et les ACL par défaut

Il y a 2 types d'ACLs : les ACLs classiques et les ACLs par défaut. Les ACLs classiques peuvent être définies sur un fichier ou un dossier et elles portent uniquement sur celui-ci. Ainsi, **elles ne sont pas transmises ou héritées** si un nouveau fichier est ajouté dans le dossier.

Les ACLs par défaut sont par contre celles **qui seront héritées** lorsqu'un fichier ou dossier sera créé. Grâce à ces ACLs par défaut, il est possible de transmettre des droits déterminés.

Ainsi, si mon dossier `testACL` dispose des ACLs suivantes :

```
user:lsw:rwx
user:swila:rwx
default:user:lsw:rwx
```

Les utilisateurs `lsw` et `swila` disposent personnellement des permissions `rwx` sur le dossier `testACL` (ils peuvent donc ajouter / supprimer / traverser le dossier). Par contre, si un dossier y est ajouté, seule la permission par défaut (`user:lsw:rwx`) sera héritée par ce nouveau dossier.

3.4.2 Manipuler les ACLs

Pour manipuler les ACLs, il y a 2 commandes principales : `getfacl` et `setfacl`. La commande `getfacl` liste les droits sur un dossier ou fichier en y incluant les ACLs qui seraient présentes :

```
$ getfacl testACL
# file: testACL/
# owner: root
# group: root
user::rwx
user:lsw:rwx
user:swila:rwx
group::r-x
mask::rwx
other::r-x
default:user::rwx
default:user:lsw:rwx
default:group::r-x
default:mask::rwx
default:other::r-x
```

Nous pouvons voir dans cet exemple plusieurs choses importantes. Tout d'abord, un rappel de l'utilisateur (`owner`) et du groupe (`group`) propriétaire. Ensuite, nous avons la liste des ACLs définie qu'il faut comprendre comme suit :

ACL	Explication
`user::rwx`	L'utilisateur propriétaire (ici `root`) dispose des autorisations `rwx` sur le dossier `testACL`
`user:lsw:rwx` **`user:swila:rwx`**	L'utilisateur renseigné (`lsw` et `swila`) dispose des autorisations `rwx` sur le dossier `testACL`
`group::r-x`	Le groupe propriétaire (ici `root`) dipose des autorisations `r-x` sur le dossier `testACL`
`other::r-x`	Les autres utilisateurs disposent des autorisations `r-x` sur le dossier `testACL`
`default:user::rwx` **`default:group::r-x`** **`default:other::r-x`**	Les ACLs *default* définissent les permissions qui seront héritées. De manière assez logique, les autorisations pour l'utilisateur propriétaire, le groupe propriétaire et les autres (i.e. les droits classiques donc) sont mentionnés
`default:user:lsw:rwx`	Cette ACL précise que si un dossier est créé dans le dossier `testACL`,

	les ACLs suivantes sont automatiquement associées :
	`user:lsw:rwx`
	`default:user:lsw:rwx`
	
	Nous remarquons, par contre, que les ACLs concernant l'utilisateur `swila` ne font l'objet d'aucun héritage (car elles ne sont pas mentionnées aussi en mode *default*).

La commande `setfacl` permet, quant à elle, de fixer les ACLs sur un fichier ou un dossier. Ainsi, à titre d'exemple, si nous souhaitons fixer les ACLs pour le dossier testACL, voici la commande :

```
$ setfacl -m u:lsw:rwx -m u:swila:rwx -m d:u:lsw:rwx testACL
```

Comme nous pouvons le voir, l'option `-m` permet de modifier ou d'ajouter des ACLs. Cette option peut être répétée autant de fois que souhaité. Ainsi, écrire `u:lsw:rwx` précise que l'utilisateur (`u`) dont le login est `lsw` doit se voir attribuer les autorisations `rwx`. Il en va de même pour l'utilisateur `swila`. Pour la dernière, `d:u:lsw:rwx` précise une permission par défaut qui sera héritée automatiquement par tous les objets enfants.

3.5 Le fichier fstab

Le système présente, dans le dossier `/etc`, un fichier texte nommé `fstab`. Ce fichier décrit toutes les partitions que le système connait et détermine les options utilisées lors du *montage*[7]de celle-ci.

Ce fichier texte est important et **toute modification doit être apportée <u>soigneusement</u>**. En effet, une erreur dans le fichier peut conduire le système à ne plus démarrer correctement.

```
#
# /etc/fstab
# Created by anaconda on Mon Jul 27 00:54:49 2015
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos-root /                       ext4    defaults        1 1
UUID=53796dc0-62e5-4048-8a2b-0dedc1425f42 /boot xfs     defaults        0 0
/dev/mapper/centos-home /home                   ext4    defaults        1 2
/dev/mapper/centos-swap swap                    swap    defaults        0 0
```

Comme nous pouvons le voir ci-dessus, le contenu du fichier `/etc/fstab` est divisé en plusieurs colonnes. Ainsi la 1ère colonne décrit **la partition concernée** : celle-ci peut être désignée par son entrée dans le répertoire `/dev` ou par un identifiant comme le UUID (identifiant unique). La 2ème colonne mentionne **le point de montage** : nous avons dans l'extrait ci-dessus `/`, `/boot`, `/home` et `swap` (qui représente la mémoire virtuelle). Cela signifie donc que lorsqu'on écrit un dossier ou un fichier dans le dossier /home, par exemple, on travaille sur une autre partition.

La 3ème colonne définit le format du système de fichier. Ainsi, les partitions `/` et `/home` sont formatées en mode *ext4* (mode courant sous Linux), tandis que la partition `/boot` est formatée en mode *xfs* (type de partition plus récent). Le format est déterminé à l'installation du système.

[7] La mise à disposition du système. Au cours de cette opération, le système vérifie la cohérence des données.

La 4ème colonne détermine les options que l'on peut préciser lors du *montage* de la partition. Nous remarquons qu'actuellement, l'option `defaults` est la seule qui est précisée. Les options suivantes sont possibles (sans que cette liste ne soit exhaustive, consulté la page de manuel de fstab pour plus de précisions) :

Option	Explication
sync **async**	Les opérations sur le système de fichiers sont faites de manière synchrone (`sync`) ou asynchrone (`async`).
auto **noauto**	La partition est (`auto`) ou n'est pas (`noauto`) *montée* automatiquement au démarrage du système.
exec **noexec**	Cette option autorise (`exec`) ou interdit (`noexec`) l'exécution de programme depuis cette partition.
suid **nosuid**	Cette option autorise (`suid`) ou interdit (`nosuid`) le prise en compte de la permission setuid[8]
rw **ro**	Cette option permet de *monter* le système en lecture seule (`ro`) ou en lecture-écriture (`rw`).
user **nouser** **users**	Cette option particulière permet de contrôler si le système de fichier correspondant peut être *monté/démonté* par un utilisateur. Ainsi, les options `user` et `users` permettent cette particularité. Par contre, l'option `nouser` réserve le droit de *monter / démonter* le système de fichier à l'administrateur (`root`) seulement.
defaults	Il s'agit d'un raccourci pour les options suivantes : `rw`, `suid`, `dev`, `exec`, `auto`, `nouser` et `async`
grpquota	Cette option active la prise en charge des *quotas groupes* collectif.
usrquota	Cette option active la prise en charge des *quotas utilisateurs*.

3.6 Les quotas

Les systèmes Linux proposent une gestion des quotas disques. Un quota est actif *sur une partition entière*. Grâce aux quotas, il est possible de limiter une ressource, qui peut être partagée entre plusieurs utilisateurs, de manière équitable.

Les quotas peuvent être définis par utilisateur ou collectivement pour un groupe (le quota s'appliquant ainsi au groupe, dans son ensemble).

Avant de pouvoir utiliser les quotas, **il est indispensable de les activer sur le système de fichier concerné**.

Ainsi, si nous souhaitons activer les quotas sur la partition `/home`, il faut éditer le fichier `fstab` pour ajouter les options suivantes :

```
/dev/mapper/centos-home /home       ext4    defaults,grpquota,usrquota        1 2
```

Ces modifications doivent être apportées prudemment.

[8] Cette permission particulière n'a pas été abordée dans le cadre de ce cours. Simplement, elle permet de changer temporairement d'utilisateur pour l'exécution d'une commande ou d'un programme. Cette particularité est utilisée pour permettre à des utilisateurs d'effectuer des tâches d'administration, par exemple. Ainsi, la commande `mount` (`/bin/mount`) utilise cette permission particulière.

3.6.1 Stratégie de mise en place des quotas

La stratégie de mise en place des quotas dans les systèmes Linux suit toujours le même schéma :

1. Activation, par modification du fichier `fstab`, du support des quotas utilisateurs et/ou de groupes
2. *Démonter* et *Remonter* la partition `/home`, afin de prendre en compte les changements réalisés dans `fstab` :
   ```
   $ umount /home
   $ mount /home
   ```
   Ou plus simplement :
   ```
   $ mount /home -o remount
   ```

   Pour contrôler que les options ont bien été prises en compte, il faut simplement exécuter :
   ```
   $ mount | grep /home
   ```
   Normalement, les options `usrquota` et `grpquota` devraient apparaître dans le résultat.

3. L'étape suivante consiste à créer les fichiers qui vont gérer les informations de quota. Ces fichiers nommés `aquota.user` et `aquota.group` sont créés automatiquement par la commande `quotacheck` :
   ```
   $ quotacheck -aucvg
   ```
   [9]

4. Il faut enfin informer le système d'exploitation que des quotas doivent être vérifiés sur le système de fichier considéré :
   ```
   $ quotaon /home
   ```

   Une fois activé, les quotas mémorisés dans les fichiers systèmes `aquota.user` et `aquota.group` sont d'application. Un utilisateur peut alors être contraint de respecter les limites qui sont imposées.

3.6.2 Edition des quotas

Une fois que le système de quota est actif, il est possible d'ajouter des quotas pour des utilisateurs ou, collectivement, pour un groupe donné. Nous allons désormais détailler ce point.

Les limites

Le quota disque limite l'espace disponible pour un utilisateur ou, collectivement, pour un groupe. Linux, dans sa gestion des quotas définit plusieurs paramètres. Ainsi, deux limites sont proposées : la *limite soft* et la *limite hard*. De plus, une période de temps, la *grace time period* est également précisée.

En fait, c'est assez simple : la *limite soft* est la limite en dessous de laquelle l'utilisateur doit, en moyenne, se trouver. La *limite hard* est la borne infranchissable définie pour l'utilisateur. La période

[9] L'option `-a` détermine que tous les systèmes de fichiers dont les quotas sont activés dans `fstab` vont être pris en compte, l'option `-u` traite les quotas utilisateurs, l'option `-g` traite les quotas groupes, l'option `-v` propose un résultat verbeux et, enfin, l'option `-c` provoque la création des fichiers nécessaires à la gestion des quotas.

grace time period définit le temps pendant lequel la *limite soft* peut être dépassée. Au-delà de ce temps, la *limite soft* se transforme en *limite hard* pour cet utilisateur.

Les *limites soft* et *hard* peuvent être définies en *nombre de blocs* (dans notre cas de 1 Ko - et donc, limite en volume -) et/ou en *nombre d'inodes* (et donc, en nombre de fichiers).

La période de temps *grace time period* par défaut est **7 jours**.

Fixer un quota utilisateur

Pour fixer un quota utilisateur, il y a 2 possibilités : le mode interactif et la ligne de commande.

Le mode interactif utilise la commande `edquota`, qui lance l'éditeur par défaut[10] :

```
$ edquota -u lsw
Disk quotas for user lsw (uid 1000):
  Filesystem              blocks       soft       hard     inodes     soft     hard
/dev/mapper/centos-home     3828          0          0        131        0        0
```

Le système nous affiche la consommation actuelle de cet utilisateur en nombre de blocs de 1 Ko (soit ici 3 828 Ko), les *limites soft* et *hard* en nombre de blocs actuellement définies (actuellement `0`). Ensuite, nous avons le nombre d'*inodes* actuellement utilisés (et donc le nombre de fichiers de cet utilisateur) et les limites configurées (actuellement `0`).

Pour modifier le quota de l'utilisateur, il faut simplement éditer les colonnes *soft* ou *hard* souhaitées et enregistrer les modifications.

Le mode en ligne de commande utilise la commande `setquota`. Cette commande est particulièrement intéressante car elle peut avantageusement être intégrée dans des scripts PERL pour fixer automatiquement les quotas des utilisateurs :

```
$ setquota -u lsw 450000 500000 0 0 /home
```

La commande demande de préciser l'utilisateur (par l'option `-u`), les *limites* en nombre de blocs *soft* 450000 Ko (ou 450 Mo environ) et *hard* 500 000 Ko (ou 500 Mo environ) et ensuite, les limites en nombre d'inodes (0 représente aucune limite). Enfin, il faut préciser le système de fichiers concerné (ici via le point de montage : `/home`).

Fixer un quota groupe

Fixer un quota groupe collectif est presque identique à celui d'un utilisateur. Précisément, nous avons toujours parlé de quota groupe collectif : cela signifie que, collectivement, tous les utilisateurs membres du groupe renseigné dans le quota sont soumis, ensemble, aux limites précisées.

Ainsi, fixer un quota en nombre de blocs de 1000 Mo pour le groupe *users* signifie que tous les utilisateurs, collectivement, peuvent enregistrer des fichiers pour un volume total d'environ 1 Go. Il est admis qu'un utilisateur consomme 900 Mo et tous les autres, le reste : il s'agit bien d'un quota collectif.

[10] L'éditeur par défaut peut être modifié par la variable d'environnement EDITOR. Il est possible également de lancer la commande edquota en modifiant temporairement cette variable comme suit :

```
$ EDITOR=gedit edquota -u lsw
```

Pour définir un quota groupe, nous pouvons ici aussi, procéder en mode interactif ou via la ligne de commande. En mode interactif :

```
$ edquota -g users
```

Ou, par la ligne de commande :

```
$ setquota -g users 1000000 5000000 0 0 /home
```

3.7 Exercices

1. Installer les quotas sur la partition `/home`

2. Préciser que
   a. Pour le compte créé pour votre voisin (voir leçons précédentes), la limite est de 250 Mo
   b. Pour les membres du groupe *etudiant* (voir leçons précédentes), la limite collective est de 500 Mo
   c. Pour l'utilisateur *bm1*, la limite est de 150 Mo

3. Vérifier que les quotas fonctionnent en copiant un large fichier dans le répertoire d'un utilisateur[11].

4. Déterminer, à l'aide de la commande `du`[12], l'espace disque occupé par chaque utilisateur.

5. Modifier le script de création des utilisateurs (voir leçons précédentes) pour inclure les quotas suivants :
   a. Chaque étudiant de 1ère année aura un quota de 150 Mo
   b. Chaque étudiant de 2ème année et 3ème année aura un quota de 200 Mo

   Vérifier, avec `edquota`, que ceux-ci sont effectivement bien configurés.

[11] Pour que cette vérification puisse se faire, il faut que ce soit l'utilisateur en question qui effectue la copie.
[12] `du` est une commande UNIX permettant de connaître l'espace consommé (*disk usage*)