Sujet 104
Disques, systèmes de fichiers Linux, arborescence de fichiers standard

Disques, système de fichier

- 104.1 Création des partitions et des systèmes de fichiers (Val. 2)
- 104.2 Maintenance de l'intégrité des systèmes de fichiers (Val. 2)
- 104.3 Montage et démontage des systèmes de fichiers (Val. 3)
- 104.4 Gestion des quotas de disque (Val. 1)
- 104.5 Gestion des permissions et de la propriété sur les fichiers (Val. 3)
- 104.6 Création et modification des liens physiques et symboliques sur les fichiers (Val. 2)
- 104.7 Recherche de fichiers et placement des fichiers aux endroits adéquats (Val. 2)

104.5
Gestion des permissions et de la propriété sur les fichiers (Val. 3)

Gestion des permissions et de la propriété sur les fichiers

- Description : Les candidats doivent être en mesure de contrôler l'accès aux fichiers en utilisant les droits d'accès et les propriétés appropriés.
- Termes, fichiers et utilitaires utilisés pour cet objectif :
  - chmod
  - umask
  - chown
  - chgrp

Droits d'accès (I)

- Le rôle d'un S.E. est aussi d'assurer la sécurité et l'accès aux données, ce qui est possible grâce au mécanisme des droits.
- Chaque fichier ou répertoire se voit attribuer des droits.
- Lors d'un accès à un ficher ou un répertoire le système vérifie si cet accès est permis ou non.

Droits d'accès (II)

- Chaque utilisateur possède un identifiant unique **UID** (User Identification).
- Chaque utilisateur est rattaché à au moins un groupe (groupe principal), chaque groupe possède un identifiant unique, **GID** (Group Identification).
- En interne, le système travaille uniquement avec les **UID** et **GID**, et pas avec les noms.
- La commande **id** permet d'obtenir ces informations.
- Les utilisateurs sont définis dans le fichier **/etc/passwd**.
- Les groupes sont définis dans **/etc/group**.

Droits d'accès (III)

- À chaque fichier sont associés un **UID** et un **GID** définissant son propriétaire et son groupe d'appartenance.
- Les droits sont affectés pour le **propriétaire**, le **groupe** d'appartenance et le **reste du monde** :
  - UID de l'utilisateur identique à l'UID défini pour le fichier. Cet utilisateur est le propriétaire du fichier.
  - Les UID sont différents : le système vérifie si l'un des GID de l'utilisateur est identique au GID du fichier. Si oui l'utilisateur appartient au groupe associé au fichier.
  - Dans les autres cas (aucune correspondance) : il s'agit du reste du monde (others), ni le propriétaire, ni un membre du groupe.

Droits d'accès (IV)

- Les droits affectés pour le **propriétaire**, le **groupe** d'appartenance et le **reste du monde** sont :
  - r : *Readable* (lecture).
  - w : *Writable* (écriture).
  - x : *Executable* (exécutable comme programme).
- 9 droits par fichier ou répertoire.
- Exemple :

```
drwxr-xr-x 29 ali lpic1 4096 Jan 14 08:42 Documents
```

Signification : Fichier ordinaire

- r : Le contenu du fichier peut être lu, chargé en mémoire, visualisé, recopié.
- w : Le contenu du fichier peut être modifié, on peut écrire dedans. La suppression du fichier n'est pas forcément liée à ce droit (voir droits sur répertoire).
- x : Le fichier peut être exécuté depuis la ligne de commande, s'il s'agit soit d'un programme binaire (compilé), soit d'un script (shell, perl, ...).

Signification : Fichier répertoire

- r : Les éléments du répertoire sont accessibles en lecture. Sans cette autorisation, la commande ls et les critères de filtre sur le répertoire et son contenu ne sont pas possibles.
- w : Les éléments du répertoire sont modifiables. Il est possible de créer, renommer et supprimer des fichiers dans ce répertoire. C'est ce droit qui contrôle l'autorisation de suppression d'un fichier.
- x : Le droit de traversée. Le répertoire peut être accédé par la commande cd. Sans cette autorisation il est impossible d'accéder au répertoire et d'agir sur son contenu qui devient verrouillé.

chmod

- Lors de sa création, un fichier ou un répertoire dispose de droits d'accès par défaut.
- La commande **chmod** (change mode) permet de modifier les droits sur un fichier ou un répertoire.
- Il existe deux méthodes pour modifier ces droits :
  - Forme symbolique ;
  - Forme numérique en base 8.
- Seul le propriétaire d'un fichier peut modifier ses droits (plus le super utilisateur).

Forme symbolique

chmod modifications chemins

modifications : {ugoa}{+-=}{rwx},

- Pour le droit de l'utilisateur, **u**, pour le droit du groupe, **g**, pour le reste du monde, **o**, et pour tous **a**.
- Pour ajouter des droits, **+**, pour en retirer, **-**, et pour ne pas tenir compte des droits précédents **=**.
- Enfin, le droit d'accès lui-même : **r**, **w** ou **x**.
- Il est possible de séparer les modifications par des virgules et cumuler plusieurs droits dans une même commande.

Forme symbolique : Exemples

```
-rw-r--r-- 1 ali lpic1 0 mar 21 22:03 file1
-rw-r--r-- 1 ali lpic1 0 mar 21 22:03 file2
-rw-r--r-- 1 ali lpic1 0 mar 21 22:03 file3
chmod g+w file1
-rw-rw-r-- 1 ali lpic1 0 mar 21 22:03 file1
chmod u=rwx,g=x,o=rw file2
-rwx--xrw- 1 ali lpic1 0 mar 21 22:03 file2
chmod o-r file3
-rw-r----- 1 ali lpic1 0 mar 21 22:03 file3
```

- Pour supprimer tous les droits, ne rien mettre après le signe =

```
chmod o= file2
-rwx--x--- 1 ali lpic1 0 mar 21 22:03 file2
```

Forme numérique

chmod modifications chemins

- À chaque droit correspond une valeur octale, positionnelle et cumulable.
- Pour encoder trois droits rwx, il faut trois bits, chacun prenant la valeur 0 ou 1 selon que le droit est absent ou présent. $2^3 = 8$, d'où une notation octale possible.
  - Le r vaut $4=2^2$, le w vaut $2=2^1$ et le x vaut $1=2^0$.

```
Propriétaire | Groupe  | Reste du monde
   r w x     |  r w x  |    r w x
   4 2 1     |  4 2 1  |    4 2 1
```

Forme numérique : Exemples

```
chmod 755 file1
chmod 644 file2
-rwxr-xr-x 1 ali lfsi3 0 mar 21 22:03 file1
-rw-r--r-- 1 ali lfsi3 0 mar 21 22:03 file2
```

- Cette modification n'est pas fine et ne permet pas de modifier un seul droit. C'est l'intégralité des droits qui est modifiée en une fois.
- La modification ne tient pas compte des anciens droits.

umask

- Par défaut, les fichiers ont les droits **666** (**rw-rw-rw-**) et les répertoires ont les droits **777** (**rwxrwxrwx**).
- Droits lors de la création :
  - **rw-r--r--** (**644**) pour un fichier
  - **rwxr-xr-x** (**755**) pour un répertoire.
- Ces valeurs sont contrôlées par un masque, lui-même modifiable par la commande **umask**. Elle prend comme paramètre une valeur octale dont chaque droit individuel sera retiré des droits d'accès par défaut.
- Le masque est le même pour l'ensemble des fichiers (022 par défaut). Il ne modifie pas les droits des fichiers existants, mais seulement ceux des nouveaux fichiers.

Calcul du masque

- Pour un fichier
  - **`rw-rw-rw- (666)`** Défaut
  - **`----w--w- (022)`** Retirer masque
  - **`rw-r--r-- (644)`** Résultat
- Pour un répertoire
  - **`rwxrwxrwx (777)`** Défaut
  - **`----w--w- (022)`** Retirer masque
  - **`rwxr-xr-x (755)`** Résultat
- Appliquer un masque n'est pas soustraire, mais supprimer des droits de ceux par défaut, droit par droit.
  - **`rw-rw-rw- (666)`** Défaut
  - **`----wxrwx (037)`** Retirer masque
  - **`rw-r----- (640)`** Résultat

Droits spéciaux (I)

- Bit SUID (*Set User ID Bit*)
  - Le programme est exécuté avec les droits de son propriétaire.
  - Applicable uniquement au propriétaire.
  - Aucun effet sans le droit d'exécution du propriétaire.
  - Changement : 4000 ou u+s
- Exemple :
  - La commande passwd et le fichier /etc/shadow.

Droits spéciaux (II)

- Bit SGID (Set Group ID Bit)
  - Le programme est exécuté avec les droits de son groupe propriétaire.
  - Applicable uniquement au groupe.
  - Aucun effet sans le droit d'exécution du groupe propriétaire.
  - Pour un fichier répertoire : les fichiers crées dans ce répertoire auront pour groupe celui du répertoire et non pas celui de l'utilisateur qui les crée.
  - Changement : 2000 ou g+s

Droits spéciaux (III)

- Sticky Bit (Set Group ID Bit)
  - N’autorise la suppression d’un fichier que par son propriétaire
  - Applicable uniquement à un répertoire.
  - Changement : 1000 ou u+t
- Exemple :
  - Le répertoire /tmp.

chown & chgrp

- Il est possible de changer le propriétaire et le groupe d'un fichier à l'aide des commandes **chown** (change owner) et **chgrp** (change group).

```
chown utilisateur chemins
```

```
chgrp groupe chemins
```

- Pour les deux commandes, les droits précédents et l'emplacement du fichier ne sont pas modifiés. Il est possible de modifier en une seule commande à la fois le propriétaire et le groupe.

```
chown utilisateur[:groupe] chemins
```

```
chown utilisateur[.groupe] chemins
```

- Seul le super utilisateur a le droit de changer le propriétaire d'un fichier. Mais un utilisateur peut changer le groupe d'un fichier s'il fait partie du nouveau groupe.