
Ligne de commande sous Linux

BTS SIO - Bloc 1 - Support et mise à disposition des services informatiques

U4 - 1.2 Répondre aux incidents et aux demandes d'assistance et d'évolution

1. Droits d'accès des répertoires et des fichiers	3
1.1. Propriété	3
1.2. Commandes chown/chgrp	3
1.3. Changer le propriétaire et groupe d'un fichier	3
1.4. Droits	3
1.5. Représentation symbolique	4
1.6. Représentation octale	4
1.7. Quelques permissions habituelles	5
1.8. Umask	5
1.9. chmod	6
1.10. Modification des droits	6
1.11. Droits étendus	7
1.12. Commande stat	7

1. Droits d'accès des répertoires et des fichiers

1.1. Propriété

Tout fichier UNIX possède un propriétaire. Au départ, le propriétaire est l'utilisateur (u) qui a créé le fichier mais "root" peut l'attribuer à un autre utilisateur. Seul le propriétaire du fichier et le super utilisateur (root) peuvent changer les droits.

Un fichier UNIX appartient aussi à un groupe (g). On définit ainsi les actions du groupe sur ce fichier. Ce groupe est souvent le groupe d'appartenance du propriétaire, mais ce n'est pas obligatoire.

On peut aussi définir ce que les autres (o) que le propriétaire ou groupe peuvent faire avec le fichier.

Rappelons que les dossiers sous UNIX sont aussi des fichiers. Les droits sur les dossiers (mais aussi les périphériques, etc.) fonctionnent exactement de la même façon que sur des fichiers ordinaires.

1.2. Commandes chown/chgrp

chown est un appel système et une commande UNIX nécessitant les droits de root pour changer le propriétaire d'un fichier ou dossier (de l'anglais change the owner).

Voici la syntaxe générale de la commande chown :

```
chown [-hLPR] [utilisateur][:groupe] cible1 [cible2 ..]
```

chgrp permet de changer le groupe d'utilisateur possédant un fichier ou un dossier. Contrairement à chown, la commande n'est pas réservée au super-utilisateur : le propriétaire peut aussi effectuer un changement de groupe s'il fait partie du groupe de destination.

Voici la syntaxe générale de la commande chgrp :

```
chgrp 'groupe' 'cible1' ['cible2' ..]
```

1.3. Changer le propriétaire et groupe d'un fichier

Par exemple attribuer l'utilisateur asterix et le groupe gaulois au fichier monfichier.txt :

```
$ touch monfichier.txt
$ ls -l monfichier.txt
-rw-r--r-- 1 roger roger 0 févr.  1 15:11 monfichier.txt

$ sudo chown asterix:gaulois monfichier.txt
$ ls -l monfichier.txt
-rw-r--r-- 1 asterix gaulois 0 févr.  1 15:11 monfichier.txt
```

1.4. Droits

À chaque fichier est associée une liste de permissions qui déterminent ce que chaque utilisateur a le droit de faire du fichier.

- La lecture (r) : on peut par exemple lire le fichier avec un logiciel. Lorsque ce droit est alloué à un dossier, il autorise l'affichage de son contenu (liste des fichiers présents à la racine de ce dossier).
- L'écriture (w) : on peut modifier le fichier et le vider de son contenu. Lorsque ce droit est alloué à un dossier, il autorise la création, la suppression et le changement de nom des fichiers qu'il contient (quels que soient les droits d'accès). Le droit spécial sticky bit permet de modifier ce comportement.
- L'exécution (x) : on peut exécuter le fichier s'il est prévu pour, c'est-à-dire si c'est un fichier exécutable. Lorsque ce droit est attribué à un dossier, il autorise l'accès (ou ouverture) au dossier.

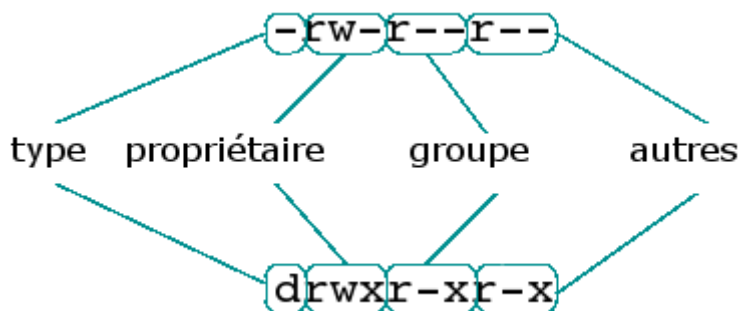
On appelle parfois r, w et x des « flags » ou « drapeaux ». Sur un fichier donné, ces 3 « flags » doivent être définis pour son propriétaire, son groupe, mais aussi les autres utilisateurs (différents du propriétaire et n'appartenant pas au groupe).

1.5. Représentation symbolique

Cet ensemble de 3 droits sur 3 entités se représente généralement de la façon suivante :

On écrit côte à côte les droits r, w puis x respectivement pour le propriétaire (u), le groupe (g) et les autres utilisateurs (o). Les codes u, g et o (u comme user, g comme group et o comme others) sont utilisés par les commandes UNIX qui permettent d'attribuer les droits et l'appartenance des fichiers. Lorsqu'un flag est attribué à une entité, on écrit ce flag (r, w ou x), et lorsqu'il n'est pas attribué, on écrit un '-'.

L'exemple suivant signifie que le propriétaire peut lire, écrire et exécuter le fichier, mais que les utilisateurs du groupe attribué au fichier ne peuvent que le lire et l'exécuter, et enfin que les autres utilisateurs ne peuvent que lire le fichier.



1.6. Représentation octale

Les valeurs octales correspondent au tableau suivant de telle sorte que les valeurs possibles pour un fichier ou un dossier sont :

Accès	---	--x	-w-	-wx	r--	r-x	rw-	rwX
Binaire	000	001	010	011	100	101	110	111
Décimal	0	1	2	3	4	5	6	7

1.7. Quelques permissions habituelles

Permission	Valeur numérique	Type de fichier	Signification
-----	000	N'importe quel type.	Aucun accès à qui que ce soit
rw-----	600	Fichiers	Fichier parano : lecture et écriture réservées au propriétaire.
rw-r--r--	644	Fichiers	Standard : tout le monde peut lire le fichier mais vous êtes le seul à pouvoir le modifier.
rw-rw-rw	666	Fichiers	Fichier public : tout le monde peut lire et écrire.
rw-x-----	700	Répertoires	Répertoire parano : lecture, accès et écriture réservés au propriétaire.
rwxr-xr-x	755	Fichiers	Fichier lisible et exécutable par tous, vous seul pouvez le modifier.
		Répertoires	Standard : tout le monde peut accéder au répertoire et lire son contenu, vous seul pouvez modifier le contenu.
rwxrwxrwx	777	Répertoires	Répertoire public : tout le monde peut y accéder, lire le contenu et modifier celui-ci.
		Fichiers	Exécutable public : tout le monde peut le lire, l'exécuter et le modifier.

1.8. Umask

Les permissions standards sont :

666 pour les fichiers

777 pour les dossiers

Umask est un masque de création de fichier qu'il faut soustraire des permissions standards pour obtenir les droits de tout nouveau fichier ou dossier créé par l'utilisateur.

Si 002 est la valeur umask par défaut :

```
$ umask 002
```

Alors les fichiers nouvellement créés auront des droits : $666 - 002 = 664$

et les dossiers auront des droits : $777 - 002 = 775$

1.9. chmod

chmod est la commande qui permet de changer les permissions des fichiers et des dossiers.

Voici sa syntaxe :

```
chmod [option] permission fichier
```

où les permissions peuvent être notées en octal :

```
$ chmod 777 fichier
```

ou en mode symbolique selon la syntaxe en utilisant :

les catégories d'utilisateur : u, g, o et a (all)

des opérateurs d'ajout/suppression : =, + et -

des droits : r, w et/ou x

```
$ chmod a+rw fichier
```

Pour assurer la récursivité, on peut appliquer les permissions à un dossier et toute son arborescence avec l'option

-R :

```
$ chmod -R u+rw nom_du_repertoire
```

1.10. Modification des droits

Créer un script rudimentaire "monscript.sh" :

```
$ touch monscript.sh
#!/bin/bash
echo "Voici mon premier script"
exit
```

```
$ ls -l monscript.sh
```

```
-rw-r--r-- 1 roger roger 50 févr.  1 15:35 monscript.sh
```

```
$ ./monscript.sh
bash: ./monscript.sh: Permission non accordée
```

Rendre le script exécutable :

```
$ chmod +x monscript.sh
```

```
$ ls -l monscript.sh
-rwxr-xr-x 1 roger roger 50 févr.  1 15:35 monscript.sh
```

```
$ ./monscript.sh
Voici mon premier script
```

N'accorder les droits qu'au seul propriétaire :

```
$ chmod 700 monscript.sh
```

1.11. Droits étendus

Il est possible d'aller plus loin dans la gestion des droits d'un dossier ou d'un fichier. Cela passe par l'utilisation des droits étendus qui offre des variantes sur l'exécution :

- SUID sur un exécutable, valeur octale : 4000, valeur symbolique : s
- SGID sur un fichier ou un dossier, Valeur octale : 2000, valeur symbolique : s
- Sticky bit, Valeur octale : 1000, valeur symbolique : t

1.12. Commande stat

La commande stat donnera des informations précises sur un fichier :

```
$ stat monfichier.txt
Fichier : monfichier.txt
  Taille : 0          Blocs : 0          Blocs d'E/S : 4096   fichier vide
Périphérique : 10305h/66309d   Inœud : 1220229     Liens : 1
Accès : (0644/-rw-r--r--)  UID : ( 1001/ asterix)  GID : ( 1002/ gaulois)
Accès : 2020-02-01 15:11:34.177590505 +0100
Modif. : 2020-02-01 15:11:34.177590505 +0100
Changt : 2020-02-01 15:13:50.172952679 +0100
Créé : -
```