

La commande ln

Lien physique
Lien symbolique

Droits d'accès :

Sous Unix il existe trois familles d'utilisateurs pour chaque fichier :

- le propriétaire
- le groupe
- les autres

et pour chacune de ses trois famille, il existe trois types d'autorisation

- r** : lecture
- w** : écriture
- x** : exécution

Le premier champ du résultat de `ls -l` (former de 10 caractères) désigne le type et les droits :

- Le premier caractère désigne le type
- Les trois suivants désignent les droits du propriétaire
- Les trois suivants désignent les droits du groupe
- Les trois derniers désignent les droits des autres

La commande chmod

`chmod`: permet de changer les droits d'accès d'un fichier donné, Seul le propriétaire d'un fichier peut modifier ses droits d'accès. Pour cela, il utilise la commande `chmod`.

1) Méthode symbolique

chmod mode nom_fichier

mode indique de quelle façon les droits d'accès doivent être modifiés. Il se décompose en **(qui) op permissions**

qui (optionnel) indique quelles classes sont concernées par la commande `chmod` et est composé de 1 ou plusieurs lettres parmi :

- u** : propriétaire (user)
- g** : groupe
- o** : autre
- a** : tous = propriétaire + groupe + autre

si aucune classe n'est spécifiée toutes les classes sont concernées.

op peut être

- +** pour ajouter des droits d'accès

- pour enlever des droits d'accès
- = pour affecter des droits

permissions = [rwxugo]

Exemple : ajouter x au propriétaire, enlever w au groupe et ajouter x au groupe. Affectez uniquement la lecture aux autres

`chmod u+x,g-w+x,o=r file`

2) Méthode octale

Les permissions du propriétaire ou du groupe ou bien des autres sont stockées sous forme de nombre octale représenté par 3 bits :

	r	w	x
Numéro du bit :	1	2	3

bit 1 : valeur 0 ou 1 (lecture)

bit 2 : valeur 0 ou 1 (écriture)

bit 3 : valeur 0 ou 1 (exécution)

On associe des valeurs numériques aux différentes permissions : **lecture : 4** , **écriture : 2**, **exécution : 1**, **pas de permission : 0**

Chaque triplet se code donc par l'addition de **4,2,1** ou **0**,

Ainsi, on a :

Valeur binaire	Valeur octale	Droits
000	0	---
001	1	--x
010	2	-w-
011	3	-wx
100	4	r--
101	5	r-x
110	6	rw-
111	7	rwX

Pour changer les droits d'accès d'un fichier :

chmod abc file

$0 \leq a,b,c \leq 7$

a désigne les droits du propriétaire

b désigne les droits du groupe

c désigne les droits des utilisateurs

Signification des droits d'accès pour les répertoires :

r: Lecture , lister le contenu du répertoire (ls)

w: Ecriture, modifier le contenu du répertoire (touch, mkdir, rm mv , cp)

x: droit de traverser (cd)

La commande Umask (Droits par défaut)

Les caractères spéciaux du Shell

Certains caractères ont une signification spéciale pour le shell :

- * Remplacé par n'importe quelle chaîne de caractères
- ? Remplacé par un seul caractère
- [...] Remplacé par un caractère entre crochet
- [!...] Remplacé par un caractère qui n'est pas entre crochet.
- \ annule la signification spéciale du caractère suivant.

La commande find :

find permet de chercher des fichiers sur le disque.

find <répertoire source> <options>

Options principales:

- name** : recherche par nom de fichier
- type** : recherche par type de fichier
- user** : recherche par propriétaire
- size** : recherche par taille
- mtime** : recherche par date de dernière modification

Exemples :

- find / -name "*.c"** : tous les fichiers se terminant par .c
- find / -size +200c** : tous les fichiers de plus de 200 caractères
- find / -size 200c** : tous les fichiers de 200 caractères
- find / -size -200c** : tous les fichiers de moins de 200 caractères
- find / -user root** : tous les fichiers appartenant à root

L'option -exec: L'option *-exec* permet d'associer une commande au résultat:

Exemple:

Afficher tous les répertoires dans l'arborescence à partir du répertoire courant

find . -type d

Ajouter le droit w aux autres pour tous les répertoires dans l'arborescence à partir du répertoire courant

find . -type d -exec chmod o+w {} \;

NB: **-type d -exec chmod o+w {}** *ici espace* **\;**

La commande chown :

'chown': permet de changer le propriétaire et/ou le groupe d'un fichier :

Syntaxe :

chown new_user:new_group file

NB: Pour des raisons de sécurité, **chown** est utilisée uniquement par **root** (L'administrateur)

COMMANDES UTILES D'UNIX/LINUX

who	Afficher la liste des utilisateurs connectés
who am i	Afficher l'utilisateur de la session courante
id	idem
date	Afficher la date
passwd	Créer ou changer de mot de passe

Redirection d'entrées/sorties sous Unix

La plupart des commandes ouvrent 3 descripteurs de fichiers qui sont assignés de la façon suivante :

- 0** : L'entrée standard (**STDIN**) associée au clavier
- 1** : La sortie standard (**STDOUT**) associée à l'écran
- 2** : La sortie standard erreur (**STDERR**) associée à l'écran

Il est possible de rediriger l'entrée et la sortie des commandes vers un fichier ou vers une autre commande à l'aide de <, >, >> et |

Redirection de la sortie standard et des erreurs

Elle est symbolisée par la syntaxe :

commande > fichier

commande >> fichier

Dans le 1^{er} cas on crée le fichier ou on l'écrit s'il existe déjà.

Dans le deuxième cas on ajoute en fin du fichier

Pour rediriger les erreurs on utilise **2>** et **2>>** à la place de > et >>

Redirection de l'entrée standard :

Elle est symbolisée par : **commande < filename**

Redirection entre commande :

La redirection entre commande est appelée **pipe** , elle est symbolisée par la caractère |
cmd1 | **cmd2** permet de rediriger la sortie de cmd1 pour servir d'entrée à cmd2

COMMANDES RELATIVES AU CONTENU DU FICHIER

cat fichier affiche le contenu de fichier

permet aussi de concaténer des fichiers. **cat fichier1 fichier2 > fichier3**

more fichier affiche le contenu de fichier page par page.

wc affiche le nombre de lignes, de mots et de caractères d'un fichier

grep: affiche les lignes des fichiers référencés contenant "Motif"

syntaxe:

grep "motif" file1 file2 ... fileN affiche les lignes des fichiers file1 file2 ... à fileN contenant "Motif"

Expressions régulières

Une **expression régulière** ou **expression** normale ou **expression** rationnelle ou motif, est une chaîne de caractères, qui décrit, selon une syntaxe précise, un ensemble de chaînes de caractères possibles.

Une expression régulière sert à identifier une chaîne de caractère répondant à un certain critère (par exemple chaîne contenant des lettres minuscules uniquement). L'avantage d'une expression régulière est qu'avec une seule commande on peut réaliser un grand nombre de tâches qui seraient fastidieuses à faire avec des commandes UNIX classiques.

Les commandes **ed**, **vi**, **ex**, **sed**, **awk**, **expr** et **grep** utilisent les expressions régulières.

^ : Début de la ligne (**^Les** -> Affiche toutes les lignes commençant par Les)

\$: Fin de la ligne (**saint\$** -> affiche toutes les lignes se terminant par **saint**)

\< : début du mot (**\<toto** -> affiche toutes les lignes contenant un mot qui commence par **toto**)

\> : fin du mot (**toto\>** -> affiche toutes les lignes contenant un mot qui se termine par **toto**)

. : remplacé par un et un seul caractère

***** : remplacé par le caractère précédent répété 0 ou plusieurs fois

[...] : remplacé par un caractère entre crochet

[^...] : remplacé par un caractère qui n'est pas entre crochet.

**** : fait perdre la signification spéciale au caractère suivant

Afficher une partie d'un fichier

tail permet d'afficher la fin d'un fichier texte

tail +20 fichier1.txt affiche le fichier à partir de la 20ième ligne.

tail -20 fichier1.txt affiche les 20 dernières lignes.

head permet d'afficher le début d'un fichier texte

head -20 fichier1.txt affiche les 20 premières lignes.

Sélectionner des données dans un fichier

cut permet d'extraire un ensemble de caractères de chaque lignes d'un fichier donné.

Les options de cut :

-c Extraction des caractères

-f Extraction des champs

-d séparateur de champs, utilisé dans le cas de l'extraction des champs.

Soient n et m deux entiers :

Extraction des caractères :

cut -cn file extrait le nième caractère de chaque ligne

cut -cn- file extrait chaque ligne à partir du nième caractère

cut -c-n file extrait les n premiers caractères de chaque ligne

cut -cn,m file extrait les caractères n et m de chaque ligne

cut -cn-m file extrait les caractères n à m de chaque ligne

Extraction des champs :

Soit **S** le séparateur de champs :

cut -dS -fn file extrait le nième champs de chaque ligne

cut -dS -fn- file extrait chaque ligne à partir du nième champs

cut -dS -f-n file extrait les n premiers champs de chaque ligne

cut -dS -fn,m file extrait les champs n et m de chaque ligne

cut -dS -fn-m file extrait les champs n à m de chaque ligne

Trier un fichier :

sort permet de trier un fichier

L'option **-t** permet de désigner le séparateur de champs (Attention : le comptage des champs commence à 0)

L'option **-n** tri par ordre numérique

sort -t : +4 -6 file trier par rapport au 5 ème champs

