

Principes généraux

Sous UNIX quasiment tout est accessible grâce à la notion de fichier : des données stockées sur le disque dur jusqu'aux périphériques de l'ordinateur. La notion de fichier est donc centrale dans le fonctionnement du système d'exploitation.

Du point de vue du cœur du système, un fichier est simplement une suite d'informations (ses *caractéristiques*) et de données non structurée (son *contenu*) stockée dans une table du système. Il est donc repérable de manière unique par son index dans cette table (son *inode*).

Le sous-système de gestion des fichiers d'UNIX permet à l'utilisateur d'avoir une vue arborescente du rangement de ceux-ci. La base de la hiérarchie de tout système UNIX est notée : «/». On appelle cela la *racine* (*root* en anglais) de la hiérarchie. Afin de pouvoir gérer cette hiérarchie il existe un type de fichier particulier : les *répertoires*.

Le contenu d'un répertoire est simplement une liste d'inode repéré par un nom. Plus simplement on dira une liste de fichiers. Un répertoire est aussi appelé un *catalogue*, ou un *dossier*. En anglais on dira plus souvent *directory*. On considère donc que ces fichiers ont la propriété de pouvoir *contenir* autant de fichiers que voulu, **y compris des répertoires**. Un fichier est donc vu par l'utilisateur comme une entrée dans un catalogue, ou encore un lien vers une inode.

Les noms de fichiers peuvent généralement comporter entre 1 et 1024 caractères. UNIX fait une différence entre les lettres en majuscules et les lettres en minuscules, de telle sorte que toto et Toto représentent deux fichiers différents.

Les fichiers dont le premier caractère du nom est le caractère «.» sont considérés comme des *fichiers cachés* : la commande `ls` ne les montrera qu'avec une option particulière.

Tous les fichiers sont identifiables par leur nom et leur emplacement dans la hiérarchie. On doit pour cela fournir la liste des répertoires qu'il faut traverser avant de parvenir dans le répertoire contenant le fichier. On appelle cette information le *chemin* (*path* en anglais) du fichier. Chaque répertoire d'un chemin est séparé du répertoire suivant par le caractère «/». Il existe deux manières de donner un chemin :

- en spécifiant la liste **la plus courte** des répertoires à traverser **depuis la racine de la hiérarchie**, on dit alors que c'est un chemin **absolu** ;
- en spécifiant **une** liste des répertoires à traverser à partir d'un répertoire particulier de la hiérarchie, on dit dans ce cas que c'est un chemin **relatif à ce répertoire de départ**.

Il existe des commandes, accessibles via le shell, permettant de se déplacer ou de modifier la hiérarchie. Le tableau 1 donne une liste de certaines de ces commandes. Les pages du manuel en ligne vont donneront des détails sur l'utilisation de ces commandes.

Utilisation

La commande `cp chemin1 chemin2` :

- crée le fichier `chemin2` si celui-ci n'existe pas,
- sinon elle écrase les données existantes dans `chemin2` par celles de `chemin1`.

Dans le premier cas, il s'agit d'une création de fichier, dans le deuxième cas, il s'agit d'une modification.

La commande `mv chemin1 chemin2` :

1. supprime l'entrée `chemin1` du répertoire dans lequel ce fichier se trouve,
2. — crée l'entrée `chemin2` avec la même inode que l'entrée `chemin1` si elle n'existe pas,
- sinon elle écrase les données existantes dans `chemin2` en les remplaçant par celles de `chemin1`.

La commande `ln chemin1 chemin2` :

- crée l'entrée `chemin2` avec la même inode que l'entrée `chemin1` si elle n'existe pas.

Commandes	Utilité
<code>pwd</code>	afficher le nom du répertoire de travail
<code>cd</code>	changer de répertoire de travail
<code>ls</code>	lister des fichiers
<code>rm</code>	supprimer un fichier
<code>mkdir</code>	créer un répertoire
<code>rmdir</code>	supprimer un répertoire
<code>cp</code>	copier un fichier
<code>mv</code>	déplacer un fichier
<code>ln</code>	créer un <i>lien</i> vers un fichier
<code>cat</code>	afficher le contenu de fichiers
<code>stat</code>	afficher les caractéristiques de fichiers

TABLE 1 – Commandes de manipulations du système de fichiers

Les droits

Chaque fichier appartient à un utilisateur, son *propriétaire*, et est rattaché à un ensemble d'utilisateur, son *groupe*. À chaque fichier sont attachés des droits d'accès qui spécifient pour chacune des catégories d'utilisateur (le propriétaire, les membres du groupe du fichier ou les autres utilisateurs) les manipulations possibles sur le fichier.

Pour chacune de ces catégories d'utilisateurs ce mode d'accès spécifie trois droits :

1. le droit de *lecture* du contenu du fichier (droit *r*)
2. le droit de *modification* du contenu du fichier (droit *w*)
3. pour les répertoires, le droit de *franchissement* du répertoire, c'est-à-dire le droit d'utiliser le répertoire dans un chemin ; pour les autres fichiers le droit d'*exécuter* le contenu du fichier, c'est-à-dire le droit de demander au système de considérer le contenu du fichier comme des instructions à exécuter (droit *x*).

Il est à noter que :

1. Pour modifier les droits d'un fichier, il faut en être le propriétaire ou être l'administrateur du système (utilisateur nommé *root*, d'UID 0).
2. Pour accéder à un fichier, il faut avoir le droit de franchissement de (*passage dans*) chacun des répertoires qui constituent son chemin.
3. Pour écrire dans un fichier, il faut avoir l'autorisation d'écriture sur ce fichier.
4. Pour créer, détruire, déplacer, renommer, ou surnommer un fichier, il faut avoir le droit d'écriture dans le **répertoire** contenant ce fichier, puisqu'il s'agit d'ajouter ou de supprimer un lien dans un répertoire (une entrée dans le catalogue).

Modification

La commande `chmod` permet de modifier les droits d'accès d'un fichier :

`chmod mode_d'accès chemins`

En version symbolique le mode est composé de la manière suivante :

<personne><action><accès>

Personne		Action	Accès	
u	propriétaire	+	r	lecture
g	groupe	-	w	écriture
o	autres	=	x	exécution/franchissement
a	tous			

En version numérique il est composé de 3 chiffres octaux (*i.e.* en base 8, donc des chiffres entre 0 et 7) représentant les droits pour :

1. le propriétaire du fichier
2. les membres du groupe du fichier
3. les autres utilisateurs

Les chiffres octaux correspondent aux droits à attribuer pour chacune de ces trois catégories d'attribution :

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

Exercices

Exercice 1 : Chemins absolus ou relatifs/Caractéristiques

Soit la hiérarchie UNIX représentée par la figure 1.

Q 1. Donnez le nom absolu du premier fichier de chaque répertoire de la hiérarchie.

Q 2. Remplissez la colonne Type de la figure 1, en spécifiant pour chaque fichier s'il s'agit d'un répertoire, d'un lien symbolique ou d'un fichier régulier.

Dans la suite de l'exercice on considère que tous les fichiers qui ne sont pas, avec certitude, des catalogues sont des fichiers réguliers. Pour simplifier on considère également que le contenu des fichiers réguliers est simplement le nom de l'animal représenté, par exemple le fichier `chacal` contient le mot `chacal`.

Q 3. Pour chacun des noms suivants spécifiez s'il correspond à un chemin relatif correct par rapport au répertoire spécifié dans la colonne.

nom	relatif à <code>/reptiles/avec_pattes</code>	relatif à <code>/oiseaux</code>
<code>../sans_pattes/cobra</code>		
<code>../../mammiferes</code>		
<code>/mammiferes/cetaces/dauphin</code>		
<code>../reptiles/avec_pattes/caiman</code>		
<code>../oiseaux/canides/chacal</code>		
<code>./caiman/../../crocodile</code>		
<code>../sans_pattes/../../oiseaux/Autruche</code>		
<code>../.</code>		
<code>/oiseaux/avec_pattes/autruche</code>		
<code>../oiseaux/autruche</code>		

Q 4. Donnez le type, l'inode et le contenu de chacun des fichiers suivants :

- `chien`
- `reptiles`
- `avec_pattes`

Exercice 2 : Manipulation du système de fichiers

Q 1. En considérant que le répertoire courant est la racine de l'arbre, que vous n'avez pas le droit d'utiliser la commande `cd` et que vous ne devez utiliser que des chemins relatifs, donnez pour chacune des phrases suivantes la (ou les) commande(s) à exécuter pour effectuer l'action demandée :

1. faire en sorte que le fichier nommé `chacal` s'appelle également `hyene` dans le répertoire `mammiferes`
2. créer un répertoire nommé `a_plumes` dans le répertoire `oiseaux`
3. effacer le fichier nommé `chacal`
4. déplacer le fichier nommé `pingouin` à la racine de l'arbre
5. copier le fichier nommé `tortue` dans le répertoire nommé `mammiferes`
6. déplacer le fichier créé par la commande précédente dans le répertoire racine en le nommant `pingouin`
7. déplacer le fichier nommé `pingouin` de la racine vers le répertoire nommé `avec_pattes` en le nommant `tortue`
8. copier le contenu du fichier nommé `loup` dans le fichier nommé `homme`.

Q 2. Donnez les numéros d'inodes de tous les fichiers de la hiérarchie après exécution de toutes les commandes de la question précédente.

Q 3. Donnez le contenu des fichiers suivants :

- `mammiferes/hyene`
- `mammiferes/singes/homme`
- `reptiles/avec_pattes/tortue`

Exercice 3 : Mode symbolique - mode octal

Soit le fichier suivant :

```
-rw-rw-r-- 1 toto iut_z Jul 7 Fich1
```

Le but de l'exercice est de déterminer les commandes permettant à l'utilisateur `toto` de modifier les droits d'accès sur le fichier `Fich1` pour avoir :

```
-rwxr-xr-x 1 toto iut_z Jul 7 Fich1
```

Q 1. Donnez une solution numérique.

Q 2. Donnez une solution avec les actions sous forme symbolique.

Q 3. La réponse à la question précédente est-elle unique ?

Q 4. Peut-on y répondre avec une seule commande en utilisant le mode symbolique ?

Exercice 4 : Validation des droits

Pour cet exercice on fait les considérations suivantes :

- les utilisateurs darwin et lamarque appartiennent au groupe serieux ;
- l'utilisateur dieu appartient au groupe blagues ;
- on se trouve dans le répertoire singes de la hiérarchie de la figure 1 ;
- on considère que le fichier hommes est un répertoire.

Q 1. Pour chacune des commandes suivantes quelles sont les droits qu'un utilisateur quelconque doit posséder pour pouvoir les exécuter ?

- cp /mammiferes/canides/loup hommes
- cp /mammiferes/singes/gorille chimpanze
- rm gorille
- ln /mammiferes/canides/loup hommes
- chmod 666 /mammiferes/singes/hommes

Q 2. Pour chacun des utilisateurs darwin, lamarque et dieu spécifiez s'ils ont le droit d'exécuter chacune des commandes précédentes.

Hiérarchie	Inode	Type	Propriétaire	Groupe	Mode d'accès
-----	-----	----	-----	-----	-----
/	2		darwin	serieux	r-xr-xr-x
-- mammiferes	3		darwin	serieux	rwxr-x---
-- canides	4		darwin	serieux	rw-xw---
-- chacal	5				
-- chien	6				
-- coyote	7				
'-- loup	8		darwin	serieux	r-x-w-rw-
-- cetaces	9				
-- baleine	10				
-- dauphin	11				
'-- orque	12				
'-- singes	13		darwin	serieux	r-x--x-wx
-- chimpanze	14		darwin	serieux	rwxr-w-r-x
-- gorille	15		dieu	blagues	-wx---rwx
-- homme	16		darwin	blagues	rwxr-x-w-
'-- orang-outan	17		lamarque	serieux	---rwx---
-- oiseaux	18				
-- autruche	19				
-- colibri	20				
-- moineau	21				
'-- pingouin	22				
'-- reptiles	23				
-- avec_pattes	24				
-- caiman	25				
-- crocodile	26				
'-- tortue	27				
'-- sans_pattes	28				
-- boa	29				
'-- cobra	30				

FIGURE 1 – **Hiérarchie animale.** Des numéros d'inodes fictifs ont été placés entre parenthèses sur la représentation pour simplifier les exercices. Le contenu de chacun des fichiers **réguliers** correspond simplement au nom du fichier en caractères minuscules.