MAJ 10/2020

M1101

UNIX : le système de fichiers

Généralités

- Bien que les icônes permettent de se passer des commandes, celles-ci sont indispensables pour comprendre le fonctionnement du système et réaliser des scripts
- UNIX est sensible à la casse !

Généralités

- Syntaxe d'une commande UNIX :

<commande> [-Ab3i] [/usr/tmp]

- Ex :

ls -ali /root

Généralités

- Le système de fichier est géré par le noyau
- Sous UNIX tout est fichier !
- Les fichiers peuvent être :
  - des fichiers de données (ou ordinaire) sur disque
  - des répertoires
  - des fichiers spéciaux pour la gestion de certaines ressources du système (par exemple, lorsqu'un périphérique d'E/S autorise des opérations d'ouverture, d'écriture, de lecture, on y accède généralement par un fichier spécial de type device)

Fichiers

- Fichiers UNIX traditionnels :
  - Les fichiers ordinaires : `-`
  - Les répertoires : `d`
  - Les fichiers spéciaux : `b c`
  - Les liens symboliques : `l`
  - Les named pipes (FIFO) communication inter-processus : `p`
  - Les sockets (réseau) : `s`

Fichiers ordinaires

- les données d'un fichier forment une suite non structurée d'octets
- caractéristiques associées :
  - date de sa création
  - date dernière modification
  - Propriétaire
  - la taille
  - etc ...
- Caractéristiques regroupées dans un descripteur de fichier, appelé noeud d'index (i-node ou index-node).

Fichiers ordinaires

Numéro i-node
type de fichier (-, d, b, c, p)
droits d'accès (rwxrwxrwx)
Nombre de liens
identifiant utilisateur
identifiant groupe
taille fichier (octets)
13 adresses de blocs de données
date dernier accès
date dernière modification
date dernière modification d'un attribut

Fichiers ordinaires

Ex : `doc1.txt`

date de création, date dernière modif …
Ceci est un petit texte

i-node : 203124

contenu

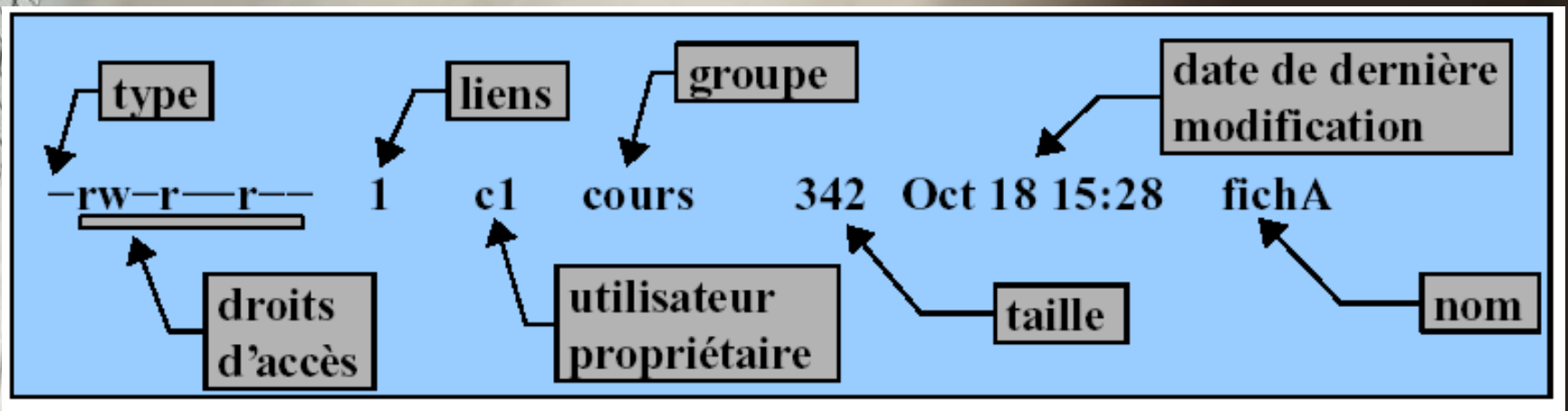
Fichiers ordinaires

- Commandes associées :

ls *fichier ...*	[**l**ist] affiche le contenu des répertoires (à un niveau) et les noms des fichiers passés en argument, c'est_à–dire *fichier ...*, ou s'il n'y a pas d'argument, tous les fichiers du répertoire courant sauf ceux commençant par un point.
cat *fichier ...*	[con**cat**enate] affiche le contenu des fichiers *fichier ...*
cp *fichier1 fichier2*	[**c**o**p**y] copie *fichier1* dans *fichier2*
mv *fichier1 fichier2*	[**m**o**v**e] renomme *fichier1* en *fichier2*
rm *fichier*	[**rm**ove] détruit le fichier *fichier* ; irréversible.

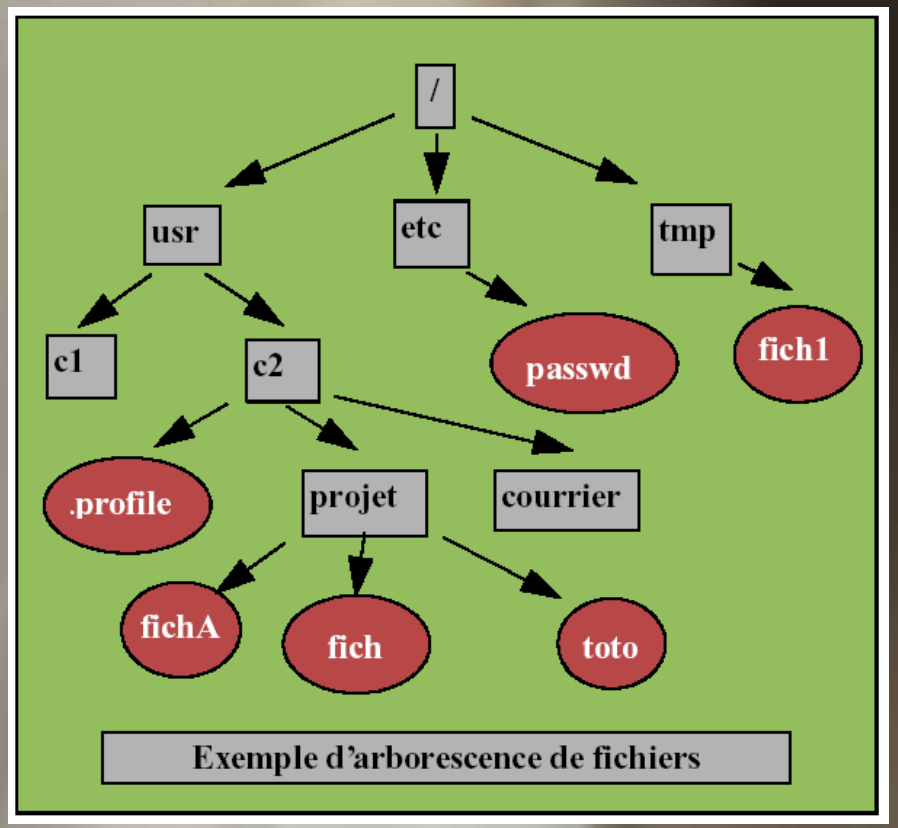
Fichiers ordinaires

- Commande `ls -l`



Répertoire

- Un **répertoire** ou catalogue (directory) est un fichier qui contient une liste de noms de fichiers, parmi lesquels on peut trouver des sous–répertoires, et ainsi de suite (arborescence logique).



Répertoires type

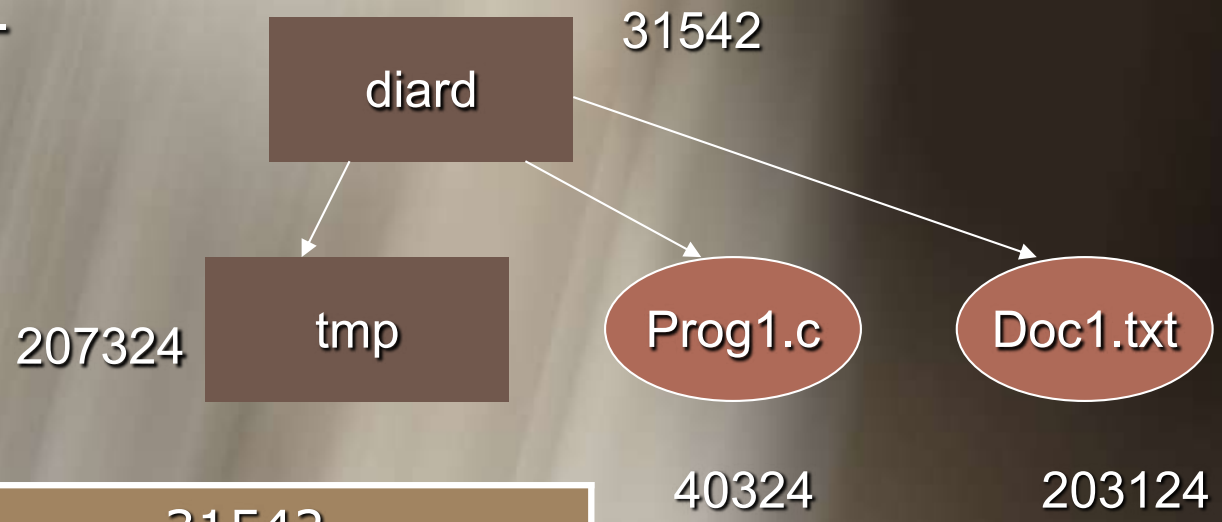
/bin	commandes UNIX de base	Sh, cat, who
/dev	fichiers spéciaux (périphs)	tty, lp, …
/etc	administration du système	passwd, group
/lib	bibliothèques objet	
/lost+found	fichiers orphelins (fsck)	
/mnt	répertoire de montage	
/tmp	fichiers temporaires	
/root	répertoire de l'administrateur	
/usr	utilitaires (man, lib, bin, tmp)	split, tr

Répertoire

- Un répertoire est un fichier ordinaire, il possède donc un i-node
- Il contient un ensemble de couples (nom, i-node)
- chaque nom est un lien vers un i−node donné.

Répertoire

- La commande **`ls -i`** permet de connaître les i-nodes
- Ex :



31542	
tmp	207324
Prog1.c	40324
Doc1.txt	203124

Répertoire

- Un répertoire est géré par le système comme un fichier ordinaire
- La création de répertoire est subordonnées aux droits d'accès
- Chaque répertoire est subordonné à un répertoire père noté : . .
- Le répertoire courant est désigné par : .
- Le répertoire de connexion est désigné par : ~

Répertoire

- Notion de chemin d'accès 2 types :
  - Chemin absolu : commence TOUJOURS par **/**
    - Ex : **`/usr/local/bin`**
  - Chemin relatif : part du répertoire courant (jamais de / au début)
    - Ex : **`user/Info1/dupont`**
    - Ex2 : **`../home/diard`**

Répertoire

- Commandes associées :

mkdir *répertoire*	[**m**ake **dir**ectory] crée un répertoire.
rmdir *répertoire*	[**rem**ove **dir**ectory] détruit le répertoire si il est vide et si ce n'est pas votre répertoire courant.
cd *répertoire*	[**c**hange **d**irectory] change de répertoire courant. Sans argument rapatrie dans le répertoire de connexion.
pwd	[**p**rint **w**orking **d**irectory] affiche le chemin absolu du répertoire courant.
ln *fichier1 fichier2*	[**l**ink] établit un nouveau lien sur le fichier *fichier1*.

Répertoire

- Commandes associées :

ls –F	affiche les noms de fichiers suivis du caractère'/' s'il s'agit d'un répertoire, et du caractère '*' s'il s'agit d'un fichier ayant la permission d'exécution.
ls –C	affiche les noms de fichiers par colonnes.
cp *fichier ... rép*	copie tous les fichiers *fichier* ... dans le répertoire *rép*.
mv *fichier ... rép*	déplace tous les fichiers *fichier* ... dans le répertoire *rép*.

Fichiers spéciaux

- Le fichiers spéciaux ne contiennent pas de données
- Ils fournissent un mécanisme qui rend les périphériques accessibles à l'aide de nom de fichier
- Ex :

Périphérique		type
/dev/cdrom	Lecteur cédérom	b
/dev/sda1	Clé USB	b
/dev/fd0	Lecteur disquette	b
/dev/ttya	Interface série	c
/dev/hda1	Partition 1 du disque hda	b
/dev/tty1	Terminal 1	c

Fichiers spéciaux

- Accès aux périphériques :
- Ex : cdrom

```
mkdir /mnt/cdrom
```

Création du répertoire de montage

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```

Format de données

périphérique

Point de montage

Fichiers spéciaux

- Accès aux périphériques :
- Ex : clé usb

```
mkdir /mnt/usb
```

Création du répertoire de montage

```
mount -t vfat /dev/sda1 /mnt/usb
```

Format de données FAT32

périphérique

Point de montage

Fichiers spéciaux

- Liste des fichiers montés :

  `mount`

- Libérer le point de montage :

  `umount /dev/cdrom`

Fichiers spéciaux

- Accès direct au périphériques :

```
cat /etc/passwd /dev/tty1
echo «bonjour» > /dev/tty1
echo «bonjour» > /dev/ttya
```

Liens

- Possibilité d'affecter plusieurs noms au même fichier (les noms différents ont le même i-node, donc correspondent aux mêmes données)
- Commande **ln**
- Syntaxe :

```
ln [-s] <original> <fichier lié>
```

- Nombre de liens sur un fichier accessible par la commande :

```
ls -il
```

Liens

- 2 possibilités :
  - Lien **dur** (hard link) :

    `ln f1 f2`

    les fichiers ont le même i-node (limité à un seul FS)
  - Lien **symbolique** (soft link) :

    `ln -s f1 f2`

    raccourci (FS différents ou non)
- Le fichier de destination d'un soft link n'existe pas forcément. Ex : cas ou le support physique n'est pas toujours disponible

Noms de fichiers

- Les noms de fichier ont une longueur <= 255 octets
- Un nom de fichier est une suite de caractères ASCII terminée par un zéro binaire
- Les nom de fichier peuvent contenir tous les caractères à l'exception de NULL \0 et /.
- Attention aux caractères de signification spéciale pour le shell (&$@-*!|?)
- Attention à la casse

Droits d'accès

- On distingue 3 modes d'accès UNIX :
- read : lire le contenu
- write : modifier le contenu
- execute : exécuter (si le code est exécutable)

ACCES	FICHIER	REPERTOIRE
r	Lire le contenu	Lire le contenu
w	Modifier le contenu	Modifier le contenu
x	exécuter	pénétrer

Classes d'accès

- 3 classes de base :
  - Utilisateur (u) droits d'accès du propriétaire
  - Groupe (g) droits d'accès pour les membres du groupe
  - Autres (others) (o) droits d'accès pour tous les autres utilisateurs du système

Manipulation des droits

- Affichage : commande **ls -l** si les droits sont présents, présence de la lettre correspondante **r**, **w** ou **x**, sinon –

```
ls -l test1

-rw-r--r-- 1 albert info1A 76 nov 7 14:23 test1
```

Modification des droits

- Symbolique :
- **+** : rajoute un droit
- **-** : retranche un droit
- **,** : opérateur de cumul

- Ex :

```
chmod u+x toto
chmod +x toto
chmod u-w toto
chmod u+rw,go-rw toto
chmod a+x toto
```

Modification des droits

- Numérique (octale) :
  - Une valeur octale = droit d'accès sur 3 bits

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

- Ex :

```
chmod 744 toto
chmod 511 toto
```

Modification des droits

- Seuls le propriétaire de l'objet et root peuvent modifier les droits de l'objet
- Différence en les deux modes de représentation :
  - Symbolique

    relatif aux droits précédents
  - Numérique

    absolu

Modification de la propriété

- On peut modifier l'appartenance des fichiers :
  - Utilisateur : **chown**

    chown user fichier
  - Groupe : **chgrp**

    chgrp groupe2 fichier
- La règle est de respecter les droits UNIX ou d'être root

Droits par défaut

- A la création de fichiers et de répertoires, assignation de droits par défaut.
- La commande **`umask`** permet de modifier les droits initiaux tel que :
- Répertoire :

  Droits effectifs = 777 - <umask>
- Fichier :

  Droits effectifs = 666 - <umask>

Niveau de sécurité

- **umask 000** → irresponsable
- **umask 077** → paranoïaque
- **umask 022** → raisonnable
- Ex :

Avec Umask = 022, les droits par défaut des répertoires sont **drwxr-xr-x**

et sur les fichiers **-rw-r--r--**

- Bonne protection :

les droits d'accès sur les fichiers doivent être cohérents avec ceux sur les répertoires