

TD Développement Web 1

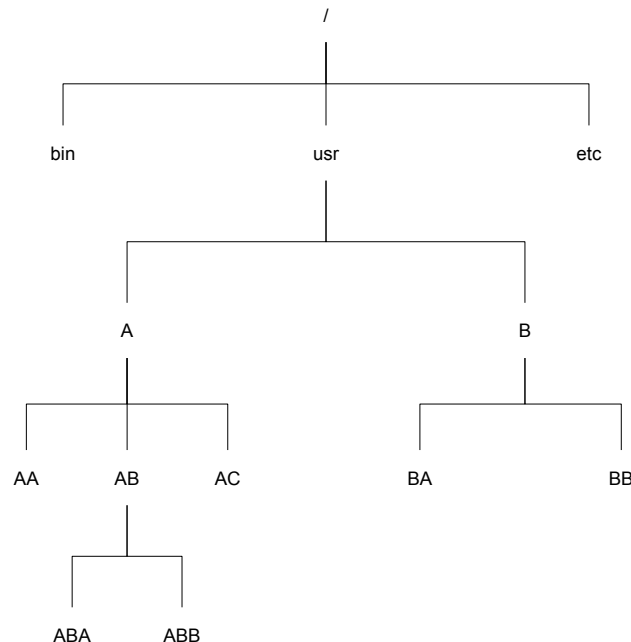
Séance 1 : Système de fichiers

Olivier Goudet, David Lesaint, André Rossi

Janvier 2019

Exercice 1

On suppose qu'on se situe dans le répertoire B de l'arborescence suivante.



1. Afficher le contenu du sous-répertoire BA en faisant apparaître les droits d'accès aux différents fichiers (2 solutions). Que représente la lettre qui précède celles décrivant les droits d'accès? Que se passe-t-il si on essaie d'afficher le contenu du répertoire BC (quel message est retourné)?

On obtient le contenu du répertoire BA avec l'une des commandes suivantes :

- `ls -l BA`
- `ls -l /usr/B/BA`
- `ls -l ./BA`

La lettre qui précède celles décrivant les droits d'accès est - pour un fichier ordinaire, d pour un répertoire, et l pour un lien symbolique.

Une tentative d'accès au répertoire BC, qui n'existe pas, se solde par le message d'erreur :

`ls: impossible d'accéder à BC: Aucun fichier ou dossier de ce type`

2. Afficher le contenu du répertoire A et de ses sous-répertoires (2 solutions).

On obtient le contenu du répertoire A et de ses sous-répertoires avec l'une des commandes suivantes :

- `ls -R ../A`
- `ls -R /usr/A`

3. Toujours en étant dans le répertoire **B**, créer un sous-répertoire **ABC** du sous-répertoire **AB** du répertoire **A** (2 solutions).

On crée le sous-répertoire **ABC** avec l'une des commandes suivantes :

- `mkdir ../A/AB/ABC`
- `mkdir /usr/A/AB/ABC`

4. Toujours en étant dans le répertoire **B**, déplacer du répertoire **A** vers le répertoire **BB** les fichiers d'extension **txt** dont le nom commence par la lettre **s** (2 solutions). Pourquoi y a-t-il deux solutions pour les questions 1 à 4 ?

On déplace les fichiers concernés avec l'une des commandes suivantes :

- `mv ../A/s*.txt BB`
- `mv /usr/A/s*.txt /usr/B/BB`

5. Comment peut-on aller dans le répertoire **etc** depuis le répertoire **B** ? Pourquoi y a-t-il deux solutions pour les questions 1 à 5 ?

On peut atteindre le répertoire **etc** depuis le répertoire **B** avec :

- `cd ../../etc`
- `cd /etc`

Il y a (au moins) deux solutions, car les répertoires peuvent être désignés par leur adresse absolue, ou par leur adresse relative.

6. Donner la commande à appliquer pour supprimer le répertoire **B** y compris ses éventuels fichiers et sous-répertoires

`rm -R /usr/B`

7. Donner la commande pour attribuer les droits en écriture sur un répertoire pour tous les utilisateurs (sans modifier les autres droits). À quels droits correspond le nombre 751 ? Expliquez en détail pourquoi.

On attribue les droits en écriture à un répertoire **repertoire** comme suit : `chmod a+w repertoire`, mais on peut aussi omettre le **a** (pour **all**) car il s'applique à tous défaut.

751 correspond à

- $7=4+2+1 = \mathbf{r+w+x}$ tous les droits pour le propriétaire
- $5=4+1 = \mathbf{r+x}$ les droits de lecture et d'exécution pour les membres du groupe
- $1 = \mathbf{x}$ les droits de traverser le répertoire pour les autres utilisateurs

Exercice 2

- Ecrire de manière symbolique (`rw-rw-rw`), les permissions suivantes : 644, 755, 711, 777, 555, 111, 600, 731

- 644 = `rw-r--r--`
- 755 = `rw-r--r--`
- 711 = `rw-r--r--`
- 777 = `rw-rw-rw`
- 555 = `r-xr-xr-x`
- 111 = `--x--x--x`
- 600 = `rw-----`
- 731 = `rw-rw-r--x`

- Que réalise la commande :

```
chmod 751 ~/.txt
```

Les droits de tous les fichiers portant l'extension `.txt` situés dans le répertoire home sont modifiés comme suit :

- Le propriétaire a tous les droits
- Les membres du groupe ont tous les droits sauf l'écriture
- Les autres utilisateurs ne peuvent qu'exécuter ces fichiers.

Exercice 3

On suppose qu'on se situe dans le home directory. Donner les commandes permettant de lister :

- les fichiers avec l'extension `.pdf`

```
ls *.pdf
```

- les fichiers se terminant par 5

```
ls *.5
```

- les fichiers commençant par L1

```
ls L1*
```

- les fichiers de 7 lettres commençant par L1

```
ls L1??????
```

- les fichiers ne commençant pas par L et dont le second caractère est un chiffre

```
ls [^L][0-9]*
```

- les fichiers contenant la chaîne `bar`

```
ls *bar*
```

- les fichiers commençant par `a` ou `A`

```
ls [aA]*
```

Copier les fichiers dont l'avant-dernier caractère est un 4 ou un 1 dans le répertoire `/home/tmp` en une seule commande.

```
cp *[14]? /home/tmp
```

Exercice 4

1. Soient les deux commandes suivantes, qu'affiche la dernière commande ?

```
a=chêne
echo "Le parc compte un \$a vert et un $a blanc"
```

Cette commande affiche `Le parc compte un $a vert et un chêne blanc`, car le `\` a inhibé le `$`

2. Donner les commande pour afficher respectivement 100 \$, `c:\`, l'ordinateur, `"salut"`, `//\`

```
echo 100 \$
echo c:\\
echo "l'ordinateur" ou encore echo l\'ordinateur
echo \"salut\"
echo //\\
```

3. Donner la valeur de la variable `a`

- (a) `a=3 ; a=$a+1`

`echo $a` renvoie 3+1

- (b) `a=3 ; a="$a+1"`

`echo $a` renvoie 3+1

- (c) `a=3 ; a=$((a+1))`

`echo $a` renvoie 4

4. Que renvoient les lignes de commande suivantes ?

- (a) `echo $(((5+6)/2))`

Cette ligne renvoie 5 (division entière de 11 par 2).

- (b) `gnome-calculator -s (5+6)/2`

Cette ligne renvoie une erreur : les parenthèses sont interprétées par **bash** alors qu'on aurait souhaité qu'elles aient un sens mathématique.

- (c) `gnome-calculator -s '(5+6)/2'`

Cette ligne renvoie 5,5.

Exercice 5

Supposons que le répertoire courant contienne les fichiers appelés **angers**, **amiens**, **avignon** et **arles**. Que retournent les commandes suivantes ?

```
echo "a*"
echo 'a*'
echo a*
```

Ces commandes affichent respectivement : **a***, **a*** et **amiens angers arles avignon**